MASTER'S DEGREE THESIS

Master of Science in Advanced Mathematics and Mathematical Engineering

Title: Fluid flow queue models for fixed-mobile network evaluation

Author: Álvaro Bernal Escribano

Advisor: Luis Velasco Esteban

Co-Advisor: Marc Ruiz Ramírez

Department: Computers Architecture

Academic year: 2018 - 2019

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master in Advanced Mathematics and Mathematical Engineering

Master's thesis

Fluid flow queue models for fixed-mobile network evaluation

Álvaro Bernal Escribano

Supervised by Luis Velasco Esteban and Marc Ruiz Ramírez

July, 2019

I would like to thank to my advisors Marc and Luis and in general all the research group GCO for making possible the opportunity of working with with top-level scientists and developing my Master thesis in exciting research projects related to networking. I express my gratitude to the GCO group for providing the academical and personal support, as well as the economic support.

Needless to say, I also thank the support from my friends from Asturias and Barcelona who have enliven this whole year with good moments. Finally, I would like to remark the unconditional support from my family, my parents and my brother who are in Asturias, and my aunts and cousins who have made my life easier here in Barcelona.

Abstract

A methodology for fast and accurate end-to-end KPI, like throughput and delay, estimation is proposed based on the service-centric traffic flow analysis and the fluid flow queuing model named CURSA-SQ. Mobile network features, like shared medium and mobility, are considered defining the models to be taken into account such as the propagation models and the fluid flow scheduling model. The developed methodology provides accurate computation of these KPIs, while performing orders of magnitude faster than discrete event simulators like ns-3. Finally, this methodology combined to its capacity for performance estimation in MPLS networks enables its application for near real-time converged fixed-mobile networks operation as it is proven in three use case scenarios.

Keywords

differential equations, queuing theory, fluid flow model, fluid queues, shared medium modelling, continuous and discrete simulation, converged fixed-mobile networks, real-time operation.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Motivation

The emergence of new services is pushing network operators to change the way network are planned and operated. Mainly, these services are high bandwidth demanding services such as Video-On-Demand (VoD) or Virtual Reality (VR), and delay sensitive services such as self driving cars or the Internet Of Things (IoT). In addition, the **dynamic nature** of the network increases due to the high number of services and consumers. Such scenarios impose enormous challenges for network operators and vendors since these services require a good **Quality of Service** (QoS) from the network.

Due to these challenges and in order to provide a good simulation tool for network operators and vendors the **CURSA-SQ methodology** was developed in [1] which solved most of these network modelling issues. The CURSA-SQ methodology provides a different point of view from the **discrete-envent simulators**, which they are based on the general assumption in network simulation literature of having M/M/1 memoryless queues. What is more, considering G/G/1 queues fits better with the behaviour of real dynamic traffic that can be extended to the traffic generated by recent network services belonging to a wide range of statistical distributions.

Note that input traffic cannot be taken directly from measuring the network traffic since most of the times is not possible because no real **monitoring data** is available for the targeted networking scenarios. That is why an **analysis of synthetic input traffic** is needed since incipient services to be supported by actual network technologies limit the availability of real monitoring data. In addition, realistic simulations needs the generation of days of traffic from thousand of devices in a network modelled as a system of queues which implies the generation of input traffic and the propagation through the queues and consequently a high execution time in general. Hence, a different approach is needed since the discrete-event simulators lacks of scalability and have a high computational cost.

The authors in [1] presented CURSA-SQ, a **fluid flow simulator** based on the **logistic queue model** which improves the Vickrey's point-queue model by allowing i) the possibility of using finite queues, ii) the possibility to obtain packet-level measurements such as delay, iii) the possibility of using practical numerical methods for solving differential equations such as ordinary differential equations (ODE) methods. On the other hand, the CURSA-SQ methodology allows to **generate input traffic** network from additional information such as the expected behaviour of the service consumers and the characteristics of the service. This approach allows to obtain much better **scalability**, the same **accuracy** and much lower **computational cost** than the discrete-event simulators being able to conduct packet and optical network planning, service introduction assessment and autonomic networking, among others.

Nevertheless, the original CURSA-SQ does not solve all the simulation challenges of the emerging networks; the main would be the convergence between **fixed** and **mobile** networks and the **real-time operation**. That is why a new extension of the CURSA-SQ is needed in order to be able to cover all these issues and the forthcoming network scenarios, including future 5G scenarios.

1.2 Contributions

As mentioned before and despite all the advantages introduced by the CURSA-SQ methodology it is not valid for mobile networks (RAN, radio access networks) and thus includes those points where converge the mobile network and the fixed network (Access-Metro networks). Since envisioned network scenarios

require an extension of the transport network towards the edge, a new part of the network must be planned and controlled. With that aim, in this Master's Thesis, the **Key Performance Indicators** (KPIs) of the network are defined allowing us to compute the end-to-end KPIs (of the entire network). These KPIs corresponds to the delay, the throughput and the traffic loss and are defined first considering one single queue and then the path between two end-points of the network. Therefore, since the original CURSA-SQ for fixed networks model is no longer valid to obtain these KPIs, an adaptation to converged fixed-mobile network scenarios must be done. As a result, a fluid flow queuing model valid for convergence fixed-mobile scenarios that can be executed for real-time network operation is obtained. In this regard, CURSA-SQ has been extended with novel models including:

- The **propagation models** to provide a model for the radio part of the RAN model.
- The model for the **scheduler** which leads to shared capacity scenarios.
- The **mobility** of the users or devices has also be taken into account. To do that, a redefinition of the traffic generator is made in terms of the service type and the system of queues topology which leads to the concept of entity.
- The **entities configuration**, proposing the optimal configuration of these entities and the initialization of all the parameters needed to run an entire simulation of the converged fixed-mobile network. Since the entities are defined in terms of the type of services we developed a **diaggregation traffic estimation** which is in charge of the service characterization of the network's traffic.

The first two models are taken into account in the **Mobile Network Model** module and the last two form the **Dynamic Configuration** module. In addition, new information from the network and the traffic has been considered in the two new DB added the **Aggregated Traffic DB** and the **Topology DB** since we add new models that need these information.

Finally, to **validate** the extended methodology we first compared it with a discrete event network simulator ns-3 [2] obtaining **high scalability** and **low computational cost** and validating the methodology as a **fast** and **accurate** way of computing the aforementioned KPIs of the network. Then, some scenarios are studied and applications from these scenarios are proposed.

1.3 Document Structure

Having stated the goals of this Master's Thesis the work will be presented in the following structure. First, all the general topics related with the radio access networks (RAN) and the fixed networks together with some insights about network management. In this background, the architecture of the RAN networks and the propagation models that have to be taken into account in every mobile communication are discussed. In addition, a brief review of the queuing theory is given in the background chapter. Then, the original CURSA-SQ model is recalled with its motivation, methodology and model.

Afterwards, we define the KPIs explaining their motivation and how to compute them. In addition, we highlight some of their applications. After seeing the necessity of a methodology able to simulate converged networks in order to compute their KPIs we propose the extensions that allow us to simulate the networks under shared capacity scenarios.

With that aim, we state how to model a RAN network to then define how each server rate is distributed among entities. In the Shared Capacity in Wireless Scenarios section, a part from describing how the scheduler works some theoretical properties and the basis of this model are explained together with some

examples. Once we know how the scheduler works we have to configure all the entities and the system of queues. To that aim, the disaggregated traffic estimation, traffic projection and entities configuration are explained. In addition, the basic model for fixed-mobile networks can be extended by considering some queues with a higher priority than others. Once the entire model have been presented we tackle three different approaches to integrate the resulting system of Ordinary Differential Equations (ODE). Finally, we end the Algorithms, methods and models chapter with some theoretical properties concerning the system of ODE in a similar way that has been done in [1].

Finally, in chapter 6 we validate the model comparing its performance with the discrete event simulator, ns-3, and provide some scenarios giving an application of the model to show its potential.

2. Background

2.1 Introduction to radio access networks

A **radio access network** (RAN) is the part of a mobile telecommunication system that implements the radio access technology. Conceptually, it resides between a device such as a mobile phone, a computer, or any remotely controlled machine and provides connection with its core network (CN) or fixed network. These networks are mainly formed by: the **eNodeBs** which has the hardware and software needed to provide the internet service to mobile phones and other wireless connected devices known as **user equipment** (UE), this terminology could change in other standards. RAN functionality is typically provided by a silicon chip residing in the core network, the base transceiver station (BTS) and the user equipments.

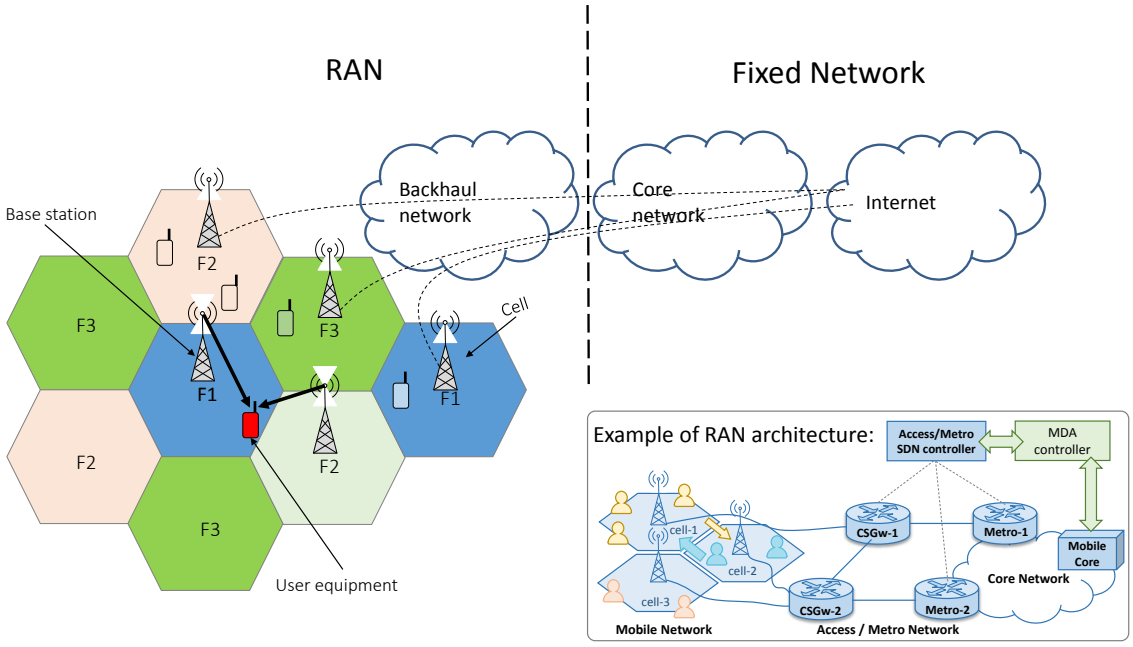

Figure 1: General architecture of converged fixed-mobile network (top) and an example (bottom).

Figure 1 shows the radio part of a RAN network with the divisions of the area into cells, each one with one frequency assigned. There can also be seen an example of a convergence of **fixed-mobile** networks where a part from the cells the RAN part includes also some gateways called CSGw and Metro nodes before the core network starts. In this work, the fixed network will be considered the core network together with the backhaul network (CSGw and the Metro routers), that corresponds to having a fixed server rate, considering mobile network only the shared medium.

Areas with high concentration of users, such as transportation stations or large commercial complexes put high stress on the BTSs that serve them. Simply adding more base stations increases the cost, and can lead to signal interference if the eNodeBs at the base stations are not carefully coordinated. Then, as the demand for connectivity has exploded, mobile operators have looked for ways to minimize the footprint and cost of their equipment. This has led to centralize some parts of the RAN and separate the functionalities of the BTS.

In fact, separating the base station into two parts, the Baseband Unit (BBU) and the Remote Radio Head (RRH), allows network operators to maintain or increase the number of network access points (RRHs), while centralizing the baseband processing functions into a *master base station*, also referred to as BBU pooling. Using a master C-RAN base station simplifies radio resource management in complex operating environments such as HetNet or Carrier Aggregation, see [3] and [4].

The resource management in RAN networks is a key factor in order provide a good Quality of Service (QoS). The **scheduler** of a certain eNodeB is the one in charge of the resources allocation for every UE attached to that specific eNodeB or Cell. The scheduler takes three main steps (QoS, channel quality and eNodeB configuration) while assigning radio resources to users, as can be seen in Figure 2. At the begining list of users connected to eNodeB is prioritized. This step allows to sort users according to their data rate requirements and retransmission needs. In the next step, the algorithm selects resources for all users by calculating their priority per each available resource block. Priority depends on queue length, packet delay, channel conditions and historical throughput.

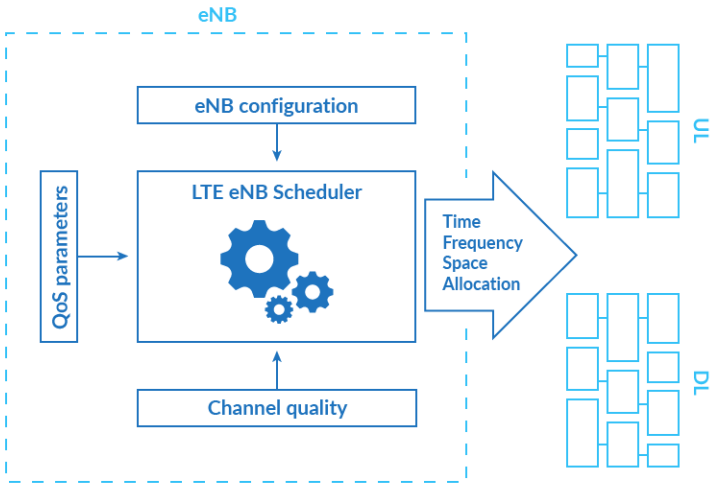

Figure 2: eNodeB Scheduler main functionalities.

There also have been some improvements in the radio part, by using new types of antennas with better aspects that we are not going to deal with. But, in order to understand how mobile communications work, we will need to explain the radio propagation model. It consists on an empirical mathematical formulation for the characterization of radio wave propagation as a function of frequency, distance and other conditions.

One of the more simple model for wave propagation is the **Friis transmission equation**, which equates the power at the terminals of a receive antenna as the product of power density of the incident wave and the effective aperture of the receiving antenna under idealized conditions given another antenna some distance away transmitting a known amount of power [5]. This model can be grow in complexity by taking into account the effect of having obstacles and interference with other cells. Once we have the propagation model for the cell we are able to compute the signal received by every device in a cell and the noise they capture. Then, we can compute **signal-to-interference-plus-noise-ratio** (SINR) in every point in the cell, i.e. the SINR perceived by a receiving antenna located at these points.

2.2 Introduction to fixed networks

In telecommunications, the **core network** is the central element of a network that provides services to customers who are connected by the access network. Typically, in telecommunication networks, the term "core" is used by service providers and refers to the high capacity communication facilities that connect primary nodes. A core/backbone network provides paths for the exchange of information between different sub-networks.

Core/backbone networks usually have a mesh topology that provides any-to-any connections among devices on the network. Many service providers would have their own core/backbone networks that are interconnected. Some large enterprises have their own core/backbone network, which are typically connected to the public networks.

The facilities and devices used for the core or backbone networks are usually routers and switches, with switches being used more often. The technologies used for the core facilities are mainly network and data link layer technologies, including asynchronous transfer mode (ATM), IP, synchronous optical networking (SONET) and dense wavelength division multiplexing (DWDM). For backbone networks used for enterprises, a 10 Gb Ethernet or gigabit Ethernet technology is also used in many instances. Core networks usually offer the following **features**:

- **Aggregation**: The top degree of aggregation can be seen in a service provider network. Next in the hierarchy within the core nodes is the distribution networks, followed by the edge networks.
- **Authentication**: Determines whether the user demanding a service from a telecom network is permitted to complete the task within the network.
- **Switching**: Determines the future span of a packet depending on the processing of the packet header, for instance Multiprotocol Label Switching (MPLS). In fact, MPLS is a routing technique in telecommunications networks that directs data from one node to the next based on short path labels rather than long network addresses, thus avoiding complex lookups in a routing table and speeding traffic flows. The labels identify virtual links (paths) between distant nodes rather than endpoints.
- **Charging**: Deals with the processing and collation of charging the data created by multiple network nodes.
- **Service Invocation**: A core network executes the service invocation task for its customers. Service invocation may occur in line with some precise activity (such as call forwarding) by the users or unconditionally (such as for call waiting).
- **Gateways**: Should be used in core network for accessing other networks. The functionality of gateways depends on the kind of network to which it is connected, for instance the Cell Site Gateways (CSGw).

In addition, the **management of the core network** has evolved through the years going from the hardware defined mainly to a more software defined scenario. In this context, it is interesting to introduce the concept of **Software-defined networking** (SDN) technology which is an approach to network management that enables dynamic, programmatically efficient network configuration in order to improve network performance and monitoring making it more like cloud computing than traditional network management. SDN is meant to address the fact that the static architecture of traditional networks is decentralized and complex while current networks require more flexibility and easy troubleshooting. SDN attempts to centralize network intelligence in one network component by disassociating the forwarding process of network

packets (data plane) from the routing process (control plane). The control plane consists of one or more controllers which are considered as the brain of SDN network where the whole intelligence is incorporated.

2.3 Queuing models

Queueing theory is the study of waiting lines. Using mathematical and computational tools a model is constructed so that queue lengths and waiting times can be predicted. In general there exist many types of queueing models, being the probabilistic the most common ones. They are very important conceptually and theoretically, nevertheless we will not doeal with them.

The second most usual models are the **discrete-event simulation** which models the operation of a system as a discrete sequence of events in time. Each event occurs at a particular instant in time and leads to a change of state in the system. In addition, no change in the system is assumed to occur between consecutive events. These simulations considers an **internal simulation clock** that rules the order of the events and allows to keep track of the simulated time. To process an event from the sequence of events, we advance the clock to the time of the event and then change the system state to reflect the impact of the event.

For the specific case of discrete event simulation for queues the system will mainly be formed by a **generator** of entities (customers, packets, ...) which follows some probability distribution (exponential, normal, ...), a **server** that serves the entities at a speed μ and a **queue** that stores the entities that could have not ben served. This means that our system can process μ entities per unit of time, if more entities arrive they will be queued. When the server gets free, the first entity in queue goes to the server and abandons the waiting line. This type of queues are called first in, first out (FIFO). Finally all entities end in a **sink** where they are terminated.

Lastly, the third ones are the continuous time queuing models or **fluid flow models** where we allow arrivals to be continuous rather than discrete. This means that instead of considering each entity discretely, we consider the continuous flow generated by them. Thus, the queue size becomes a continuous function of time. Note that discrete event simulation contrast with continuous simulation in which the system state is changed continuously over time on the basis of a set of differential equations defining the rates of change of state variables. An example of these fluid flow models would be the logistic queue model, which is the one used in the present work.

2.4 Summary

In this chapter we have given the main concepts about RAN networks and fixed or core networks and how they are managed. The key ideas is that RAN needs a wave propagation model, given by the physical media, and a model for the scheduler, given by the architecture of the RAN and its configuration. At the end, the aim of both models is to provide a characterization of the network, which can be seen as a system of queues, in order to ensure a certain QoS and thus implies ensuring a boundness in the KPIs of every user/consumer group. That is why in the nest chapters we will make a deeper look into the definition of the KPIs (Chapter 4) and the model of the RAN and the Scheduler (Chapter 5). But, before that, we need to recall the logistic queue model and the service-based traffic model which will be the basis of the proposed models in this Master's Thesis. This recall could have been explained as part of the background chapter but as it is self-contained a specific chapter has given to it, which is the following one. Finally, we end up with a brief explanation about the main concepts concerning the discrete event simulators and the continuous simulators applied to queuing theory.

3. CURSA-SQ for service-centric traffic analysis

3.1 Introduction

As we have mentioned before discrete-event simulators have scalability issues since they are not able to reproduce a real scenarios, which consist on traffic of many consumers during several days and through a network formed by a system of queues, without expending more time in execution than the actual simulation time. That is why, in order to have a better scalability and time efficiency, continuous queue models can be proposed like Vickrey's point-queue. However other issues may appear, such as the restriction of using infinite queue, the impossibility of obtaining packet-level measurements such as delay, and the impossibility to use practical numerical methods for solving differential equations. That is why, the authors of the paper [1] proposed a traffic flow analysis methodology in addition to a traffic flow generation (based on service characteristics) in order to avoid the lack of monitoring data.

We begin by explaining the general idea of this service-centric traffic flow analysis methodology behind the logistic queue model and then we will explain the queue model, considering we have a continuous differentiable input traffic grouped in services. Moreover, this methodology will be extended for heterogeneous networks with a wireless part and a fixed part, like the Access-Metro networks.

3.2 Methodology

In this section, a fast, accurate, attainable and scalable service-centric traffic flow analysis methodology [1] is described. This methodology solves the main issues of the discrete-event simulators and the main issues of other continuous queue models mentioned through continuous queue models and statistical flow characterization. The key idea of this methodology is to consider a communication between service consumers and service providers which can be extended to other scenarios like function-to-functions in virtualized network functions networks, etc. In this service-to-service communication there will be an upstream flow of traffic that arrives from service consumers in a network node that aggregates and fordward it towards the service provider, dowsntream flow of traffic that comes form the service provider to the service consumer in response to the service requests, a node capacity k given by the capacity of the buffers and a link capacity μ, see Figure 3. Moreover, we will group the users with the same traffic characteristics in order to reduce the number of input traffic flows. Then, we will distinguish two steps: the first one which consists on the generation of each user traffic based on the service characteristics and the aggregation of these users into consumer groups of a certain service type and a second step which is the queue system and the queue model which will receive the input traffic generated in the first step.

For the first step, different traffic flow generators are considered with different bitrate trace evolution and different granularities T, in general sub-second granularity. Because of the integration process for solving the ODE system is necessary to reproduce the packet-based traffic and its characteristics with a traffic flow generator characterized by a bitrate trace along time, see 2.a) of Figure 3. In order to have a good approximation between both generators different parameters attending to the service characteristics of each consumer group will be configured. These consumer group characteristics are:

- Consumer behaviour: the behaviour of the consumers of a specific service, can be modelled by characteristics which depends on the service, for instance for a VOD consumer are te time between consecutive reproductions, duration of the content, etc.
- Data exchange: how the data is going to be transfer according to the consumers' activity. Continuing

with the same example, in VOD the data exchange will be given by the buffer size and the ON/OFF pattern the media segments follow in the data exchange process.

- Consumer infrastructure: adaptation between the data exchange and the packet network because the network can impact in the service, i.e. this could impact in the size of the media segments and consequently, on the packet traffic characteristics.

Then, these consumer group characteristics can be quantified by random variables which represent the contribution of each user has in the group taking into account the packet-based nature networks have, see 2.b) of Figure 3. So, every users' traffic will be characterized, on one hand, by the inter-arrival burst rate and the burst size for the traffic flow characterization and, on the other hand, by the inter arrival packet rate and the packet size for the packet-based traffic characterization.

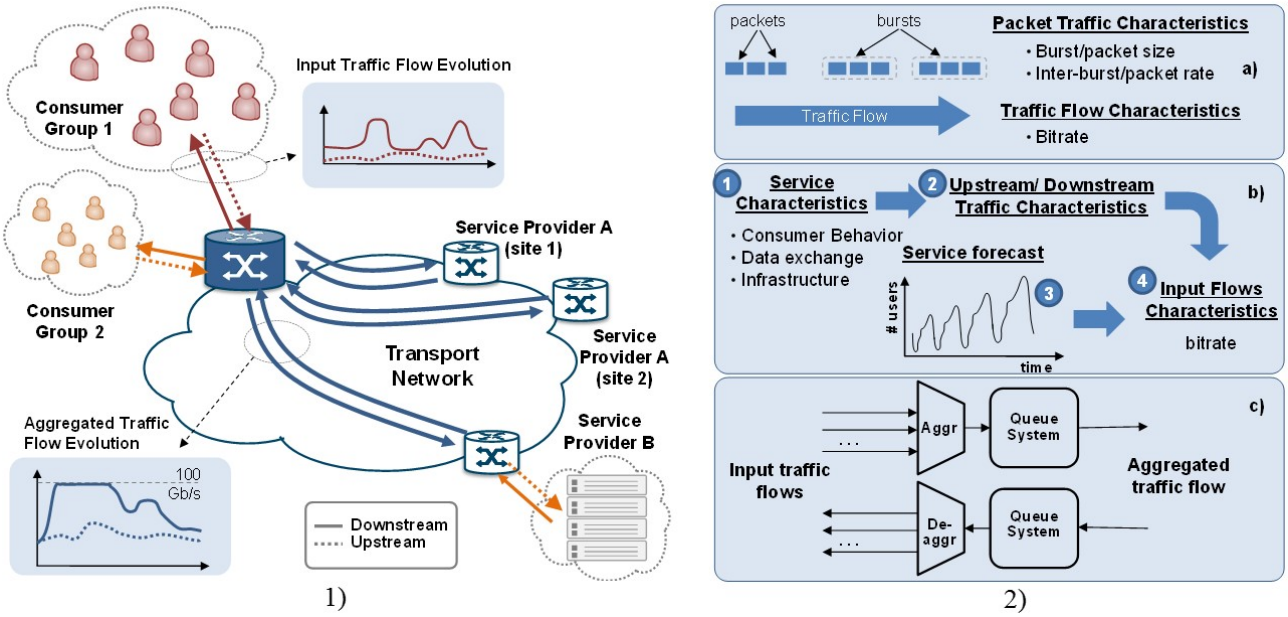

Figure 3: 1) General Overview of targeted scenarios, 2) Overview of CURSA-SQ Methodology.

Once the bitrate generated by every user of a given consumer group is computed for the upstream, they are aggregated in terms of the same service characteristics and we obtain the traffic flow generated by one consumer group. Each consumer group traffic flow is introduced in their corresponding queue system and fordwarded as appear in 2.c) of Figure 3. In the downstream the precedure is analogous except the fact that we start having the aggregated traffic flow and disaggregate at the end of the process.

3.3 Queue system and model and input traffic charactrization

The network is formed by queuing modules which consists on an input traffic measured in bps, a First-In-First-Out (FIFO) queue (Q) with capacity k bytes and an output measured in bps and limitted by the server-bitrate μ which is the rate at which the server process the data coming from a consumer group (it can also represent the link capacity). Therefore, we can build an entire network using these queue modules and the correspondent aggregators.

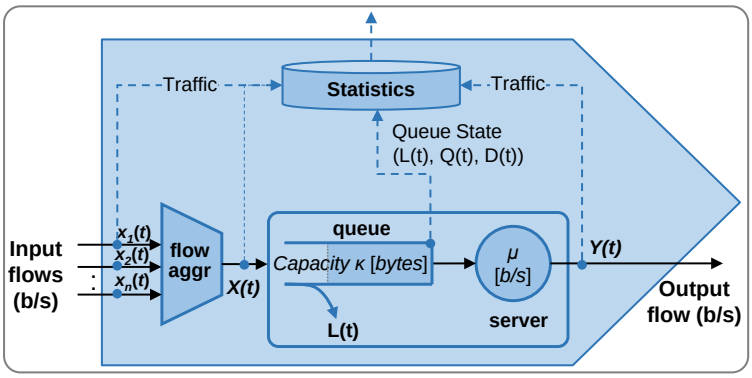

Figure 4: CURSA-SQ queuing module.

In the logistic queue model of Figure 4 n continuous input traffic flows are aggregated into a single input flow $X(t)$, then queued in the capacitated queuing system, which finally leaves the queue as the output traffic flow $Y(t)$. Since the queue has a limited capacity, traffic loss $L(t)$ may appear if the input traffic flow exceeds the available capacity in the queue at time t. That is why $\hat{X}(Q(t), t)$ is defined as the traffic flow that actually enters in the queue at time t. The queued data remains for a time $D(t)$ in the queue until the server in the queue module processes them at a constant rate μ.

Logistic queue model

Let us consider a well defined input bitrate (b/s) $X(t)$ at time t in the time interval $t \in [t_0, t_1]$. This input can be a piece-wise linear interpolation from discrete data with granularity T. On one hand, by the flow conservation equality, we can stablish

$$Q(t + \Delta t) = Q(t) + \frac{1}{8} \cdot (X(t) - Y(t)) \cdot \Delta t, \tag{1}$$

where $Y(t)$ is the output bitrate (b/s) at time t and Δt is a small time interval ($<< T$). By taking limits we obtain the differential equation

$$Q'(t) = \frac{dQ(t)}{dt} = \frac{1}{8} \cdot (X(t) - Y(t)). \tag{2}$$

On the other hand, the output bitrate can be expressed in terms of the input bitrate by using the logistic queue model,

$$Y(t) = \mu + (min\{\mu, X(t)\} - \mu) \cdot e^{-8\lambda \cdot \frac{Q(t)}{\mu}}, \tag{3}$$

where μ is the server bitrate (in b/s), $Q(t)$ is the backlogged traffic (in Bytes) at time t and λ controls the emptying slope of the queue, so it should be proportional to the input traffic (in our case we set it as $\lambda \sim avg(X/\mu)$).

In addition, to add the capacitated queue restriction we have to reformulate the input traffic as,

$$\hat{X}(Q(t), t) = \frac{X(t)}{1 + h \cdot e^{\rho \cdot (Q(t) - k)}}, \tag{4}$$

where k is the capacity of the queue and h a weighting coefficient which models how much effect does have the packet drops in terms of the relation between the input traffic and the server-bitrate, e.g. $h =$

$-(1 - max(X/\mu))$. Note that every adimensional parameter which is needed to model the queue is adimentionalized by normalizing by μ, like we do in the emptying slope or in the packet drops importance. On the other hand, the ρ has to be tuned in order to adjust properly the slope of the descend when the queue is almost full taking into account the ODE convergence of the integrator (fast descend but with enough smooth function).

Therefore, plugging equations 3 and 4 into 2, we end up having,

$$Q'(t) = \frac{1}{8} \cdot \left[\hat{X}(Q(t), t) - \left[\mu + \left(min\left\{\mu, \hat{X}(Q(t), t)\right\} - \mu\right) \cdot e^{\rho \cdot (Q(t) - k)}\right]\right]. \tag{5}$$

Then, having an initial state of the queue, Q_0, at time t_0 and a final integration time, t_{max}, we can compute the state of the queue at any time t, with $t \in [t_0, t_{max}]$, just by solving the differential equation 5. In addition, it is very usual to have more than one queue and more than one stage of queues in real scenarios. The procedure in these scenarios will be to identify the different stages which conforms the system of queues and integrate equation 5 with the input of the previous stage or the initial conditions obtaining the result of the following stage, forward-propagating the traffic.

Finally, if no real input traffic traces are available a characterization of the input traffic can be made taking into account the consumer behaviour, the data exchange and consumer infrastructure. These characteristics can be translated into four random variables: packet size, burst size, inter-arrival packet rate and burst inter-arrival rate.

Moreover, from the perspective of a flow aggregating several individual active consumers, the effect of both packet size and inter-arrival packet rate variables can be neglected compared to burst size and burst inter-arrival rate. Then, an expression of the input traffic can be given in terms of the following parameters:

- *ibr* Inter-arrival burst rate (s^{-1}), defined as the rate of consecutive bursts.
- *bs* Burst size (in bits for convenience)
- r Consumer maximum flow rate (bps)
- γ Traffic burstiness degree
- $U(t)$ Number of active consumers at time t
- $X(t)$ Bitrate (bps) generated by a consumer group or service provider site
- T Traffic generation granularity (s)

Then, the input traffic for second granularity can be obtained as,

$$X'(t) = min\{U(t) \cdot r, \Phi(E(X(t)), V(X(t)))\} \tag{6}$$

where *Phi* is a given distribution (e.g. uniform or Gaussian) and $U(t) \cdot r$ is the maximum traffic that the consumer group can inject/receive due to access speed constraints. In addition, $E(X(t)) \approx E(U(t)) \cdot E(bs \cdot ibr) = E(U(t)) \cdot E(bs) \cdot E(ibr)$ due to independence between the random variables, and $V(X(t)) \approx E(U(t)) \cdot V(bs \cdot ibr)$.

For sub-second granularity we can extend the equation 6 obtaining the following equation in terms of the burstiness degree that can be interpreted as a duty cycle.

$$X''(t,i) = \begin{cases} min(U(t) \cdot r, \gamma^{-1} \cdot X'(t)) & \text{if } T \cdot \sum_{j=0\ldots i} X''(t,j) < X'(t), \\ 0 & \text{if } T \cdot \sum_{j=0\ldots i} X''(t,j) \geq X'(t) \end{cases} \tag{7}$$

where γ has an impact on the number and magnitude of samples in the on period and is computed as

$$\gamma = \frac{bs/r}{bs/r + 1/ibr}.$$

Finally, it is worth noting that, if $T > 1$ second, $X'(t)$ can be easily computed by averaging random samples generated with 1 second granularity, whereas $X''(t,i)$ do not need to be computed.

3.4 Conclusions

An alternative to discrete event simulators can be proposed, such as the CURSA-SQ methodology which provides a generation and analysis of the service-based traffic flows. To do that accurately, the generated synthetic traffic flows are based on service characteristics and consumers behavior allowing us to analyze the impact of the service on the network infrastructure. In addition, the continuous queue model from [1] was formally recalled, which is a key component to create queuing systems.

Moreover, the CURSA-SQ methodology shows high accuracy and extraordinary scalability compared to the traditional packet-based generation and simulation, the numerical validation of this statement can seen in [1]. Thus, the CURSA-SQ methodology can be used in applications such as packet and optical network planning, service introduction assessment and autonomic networking; but, extensions need to be carried out if real-time operation in converged fixed-mobile networks is wanted. This new use case and the extensions to the original CURSA-SQ methodology explained in this chapter will be the next topic to be presented.

4. KPIs for Fixed-Mobile Network Real-Time Operation

4.1 Introduction

In the previous section, we have seen a methodology that enables to do the planning of a network, do service introduction assessment and autonomic networking for fixed (optical) networks. But, as we move into actual scenarios and its stringent requirements where there is a major convergence between mobile-fixed networks and these networks are more dense, the transport network needs to be extended towards the edge, implying a real-time characterization of the KPIs [6]. Consequently, a new use case appears which is the real-time operation for fixed-mobile network where KPIs are introduced in order to have reliable measures of how our system (the Fixed-Mobile Network) evolves.

Therefore, in this section is introduced the general idea and the needs for fast and accurate end-to-end KPI estimation, like throughput, delay and lost bytes. Mobile network features, like shared medium and mobility, are presented, which combined to its capacity for performance estimation in MPLS networks enables its application for real-time converged fixed-mobile networks operation.

Thus, the scope is to extend the CURSA-SQ methodology in order to determine the KPI of a given network in real-time with application to Autonomic Networking in 4G and 5G scenarios. As there is a convergence between mobile and fixed networks it is necessary to analyze the users' mobility. As a result of users mobility, large traffic variations can be expected, which might cause congestion thus increasing end-to-end delay and bytes losses and decreasing throughput. As a consequence, real time operation to manage resource utilization and to adapt them to current and near-term network conditions is strictly needed.

4.2 KPIs for Real-Time Operation

In Figure 5a is presented a general scenario of the convergence between mobile and core networks through the CSGw which connects a base station in the mobile network to a set of packet nodes in the fixed metro network. An SDN controller is considered to be in charge of the access-metro network (formed by CSGws and packet nodes in a geography). At one end of the network, the user activity on a number of heterogeneous services (video streaming, P2P, gaming, etc.), dynamically injects traffic into the fixed network. Note that: i) the number of active users in a given cell fluctuates not only at macroscopic scale (daily) but also at microscopic one (sub-second) according to complex behavioral aspects [7]; ii) the traffic generated by the different services is not constant, as non-deterministic on/off patterns are commonly observed [8]; and iii) users mobility, including within the same cell and among neighboring cells, impacts on the end-to-end latency and throughput in both upstream and downstream directions [9]. For these reasons, one can easily conclude that pursuing the optimum operation oriented to guarantee end-to-end KPIs with efficient resource usage is a complex task requiring complex control extending typical SDN controller capabilities.

Therefore, the KPIs will allow us to determine the behaviour of the network as a system of queues modelled by the logistic queue model with adaptations to converged mobile-fixed networks. First of all, the physical and functional characteristics of the cell and the users in it must be parametrized. To do that, the users' positions and service-based traffic have to be forecasted by machine learning (ML) algorithms for instance. Then, knowing the basic characteristics of the cell (diameter and SINR mainly) its topology can

be build having different zones (in terms of the SINR) generating the different types of considered traffic in each region. Note that, the physical information of the RAN and the different regions considered to discretize the cell are stored in the Topology Data Base (DB).

Thanks to the discretization of the cell we are able to aggregate in each region the traffic flows from different devices into consumer groups in terms of the service characteristics, following the same philosophy explained in the original CURSA-SQ methodology. This proposition lead us to have only N flows, each one associated to one consumer group characterized by the type of service (Services DB) and the region where is located (Topology DB). In addition, each consumer group injects traffic proportional to the number of users contained in the consumer group at a certain instant of time t, thus all the CURSA-SQ parameters of the model will be affected accordingly as can be seen with the queue size in Figure 5b.

Once we have the correspondent percentage of traffic to each service (Aggregation Traffic DB) and the Topology, synthetic traffic flow is generated and propagated to the correspondent system of queues allowing us to analyze the KPIs of flow i from the queued traffic (delay), $Q_i(t)$, throughput $Y_i(t)$ and lost bytes, $L_i(t)$, as is shown in Figure 5a down. For instance, in Figure 5a users' group 1 moves from cell-1 to cell-2 leading to a decrease of the throughput, i.e. increase of delay, in cell-2 and a decrease of throughput, i.e. increase of delay, in cell-1 due to the congestion of the cell-2. In order to solve that issue, we can think of providing more bitrate to link 3 in regard of decreasing birate to links 1 and 2. Different configuration can be subsequently simulated to find the ones that minimizes or avoid the detected congestion. Finally, CURSA-SQ produces outputs to the decision maker in charge of managing resource allocation, as well as traffic engineering policies. In order to accomplish this traffic behaviour characterization in real-time converged fixed-mobile networks two new modules and two Data Bases (DB) have been added to the CURSA-SQ methodology, Figure 5c.

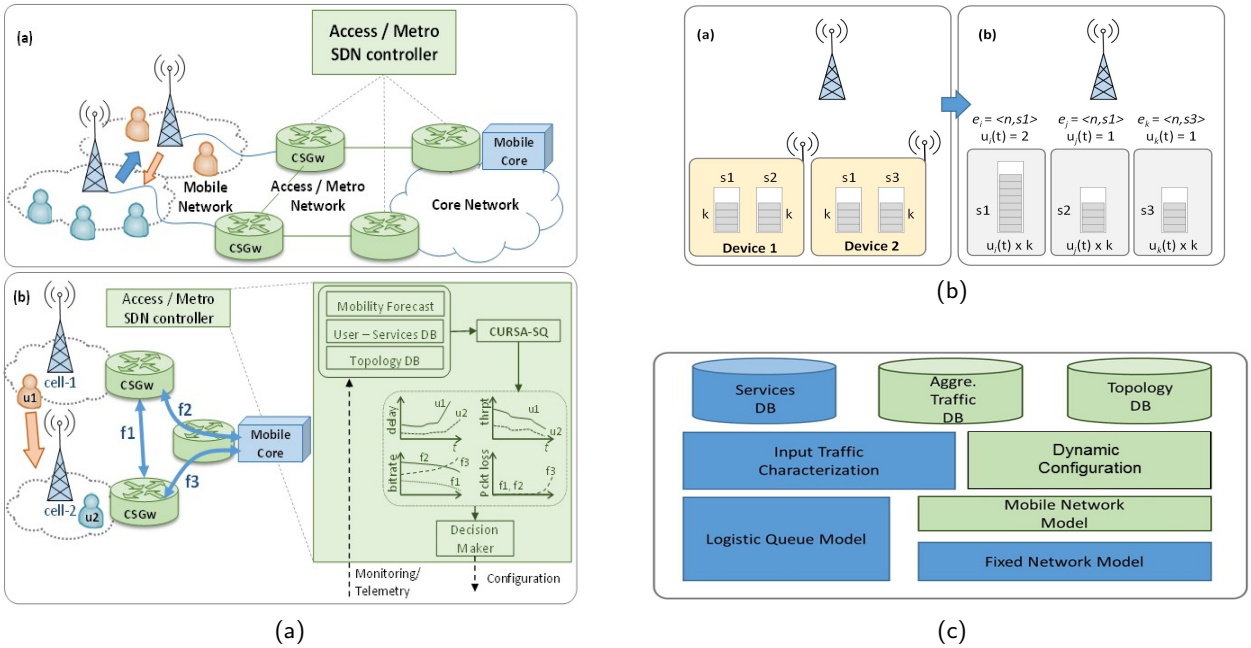

Figure 5: (a) Converged fixed-mobile network (b) real-time KPI estimation (c) CURSA-SQ modules (in green the added ones to model RAN).

After computing the outputs generated by the CURSA-SQ simulation, given its inputs (the system of queues topology and input traffic in each flow) and parameters (queue capacity, rate sharing, ...), the KPIs

per queue are obtained (throughput, delay and traffic loss). So, let us define these parameters which will characterize the system of queues and allow us to make decisions about the network configuration, among other things. The first one will be directly the output traffic exiting the queue associated to a given queue $q \in \mathcal{Q}$, being $\mathcal{Q}$ the set of queues that form the complete network.

Definition 4.1. Throughput, $T(t)$ or $Y(t)$. The volume of bits which flows through one entity formed by a generator, a finite queue and a server rate. This throughput will always be bounded between zero and the server bitrate, i.e. is a bounded non-negative real function. It can also be defined the relative throughput which is the proportion between the output bitrate and the input bitrate of a given queue.

The second indicator will measure the waiting time of a user at the end of the network (end-to-end delay), which can be computed from the delay introduced by each queue between the two ends.

Definition 4.2. Delay, $D(t)$. In the discrete model the delay can be defined as the time a packet has to wait in order to exit the queue when it has entered the queue. The idea for continuous models is similar but considering flows instead of packets. Having the evolution of the backlogged traffic, $Q(t)$, the delay perceived by the input traffic flow at instant t can be computed as,

$$D(t) = 8\frac{Q(t)}{\mu(t)}, \tag{8}$$

which will be equivalent to the delay of a packet that has entered into the queue at a certain instant of time t and where $Q(t)$ is the backloged traffic at time t and $\mu(t)$ is the server bitrate at time t. Remark that the actual time needed by a packet, that enters at time t, to exit the queue does not correspond to the delay perceived by this packet at that certain time t because the server bitrate will not always be the same along time.

Lastly, the third parameter will measure all those bytes lost in the queuing process. In addition, with these parameters we can check the conservation law since $X(t) = L(t) + Q(t) + Y(t)$.

Definition 4.3. Losses, $L(t)$. In discrete model packet losses are usually defined as the packets which can be neither transmitted neither backlogged in the queue because the queue capacity has been reached. In the proposed continuous model the lost bytes can be easily computed from the formulation of the finite queue subtracting equation 4 from the input traffic, $X(t)$. Moreover, $\hat{X}(Q(t), t)$ is computed from a step function in terms of the queue capacity, k, times the input traffic. So, the lost bytes are a bounded non-negative real function, i.e. $L(t) = X(t) - \hat{X}(Q(t), t) \in [0, X(t)]$. For convenience and simplicity, the definition of losses used in further steps will be as a percentage of the input traffic,

$$L(t) = 1 - \frac{\hat{X}(t)}{X(t)} \in [0, 1]. \tag{9}$$

Let us now consider we have $n = 1, ..., N$ entities at one end of the network, each one associated to one specific region and type of service for the RAN network, $s = 1, ..., S$ services providers at the other end of the network, and in between the mobile/fixed network. Then, each entity will have a certain achievable rate given by the SINR entity n experience. Moreover, the RAN or mobile network injects traffic in the Access/Metro network and this, in turn, to the Metro Core Network; both fixed networks with a given planar topology, $\mathcal{P}$. Therefore, the end-to-end KPIs are computed from the KPIs of the RAN part and the KPIs of the fixed network. Both parts will be formed by a system of queues, each one with its KPIs previously defined.

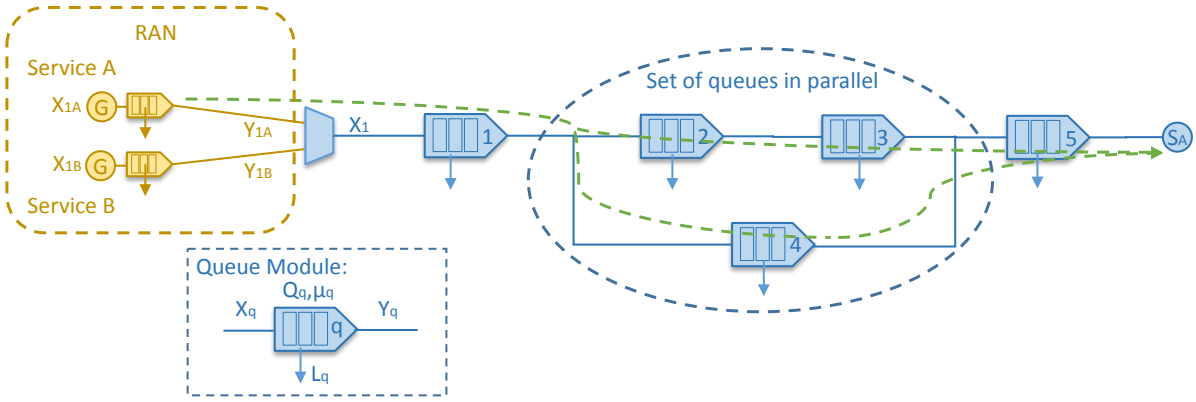

Figure 6: End-to-end KPIs example.

On one hand, the results obtained from the continuous simulation of the RAN part will allow us to determine the KPIs per-entity, which corresponds with the KPIs of the queue. Consequently, per-device partial KPIs are computed proportionally from per-entity ones. In other words, per-device partial KPIs are computed simply dividing by the number of users contained in an entity. This can be made without any loss of the accuracy because the aggregation of users into entities have been made concerning similar traffic generation behaviours and hardly separated (see Section 5). On the other hand, in the fixed network, there will be aggregations and disaggregations of flows which means that the KPIs will also depend on the topology of the network. The topology of the network can be sliced into blocks, each one with a single queue or a set of queues disposed in parallel. Then, per-device KPIs are computed from per-device partial KPIs (contribution of the RAN network into the KPIs) and the contribution to the KPIs of the fixed network. For instance, the end-to-end throughput for one path (i.e. one device), $p \in \mathcal{P}$, will be

$$T_p(t) = \left[(1 - L_{RAN}(t)) \cdot \prod_{i \in \mathcal{I}} (1 - L_i(t)) \cdot \prod_{j \in \mathcal{J}} (1 - L_j(t)) \right] \cdot X_p(t) \tag{10}$$

where $\mathcal{I}$ is the set of single queues and $\mathcal{J}$ is the set of all groups of parallel queues in the topology, as can be seen in the example of Figure 6. In turn, the throughput of the parallel queues is computed taking into account the aggregation percentages, α_b with $b \in \mathcal{B} \subseteq \mathcal{J}$ the set of branches of one set of queues in parallel and $\mathcal{K} \subseteq \mathcal{B}$ the set of queues of one branch, as

$$(1 - L_j(t)) = \sum_{b \in \mathcal{B}} \alpha_b \left[\prod_{k \in \mathcal{K}} (1 - L_{bk}(t)) \right].$$

For instance, in the example of Figure 6 there are 2 branches ($|\mathcal{B}| = 2$), the first one with 2 queues ($|\mathcal{K}| = 2$) and the second one with one ($|\mathcal{K}| = 1$). The definition for the end-to-end delay for one path will be made following the same criteria and making use of the same sets, being defined as

$$D_p(t) = d_{RAN}(t) + \sum_{i \in \mathcal{I}} d_i(t) + \sum_{j \in \mathcal{J}} d_j(t) \tag{11}$$

where the delay introduced by the set with parallel queues is

$$d_j(t) = \max_{b \in \mathcal{B}} \left\{ \sum_{k \in \mathcal{K}} d_{bk}(t) \right\}.$$

For the traffic loss (in %) the definition is exactly the same as the throughput but taking into account that $T(t) = (1 - L(t)) \cdot X(t)$,

$$L_p(t) = L_{RAN}(t) \cdot \prod_{i \in \mathcal{I}} L_i(t) \cdot \prod_{j \in \mathcal{J}} L_j(t) \tag{12}$$

$$L_j(t) = \sum_{b \in \mathcal{B}} \alpha_b \left[\prod_{k \in \mathcal{K}} L_{bk}(t) \right].$$

Finally, per-device partial KPIs will be useful to determine the state of the RAN network, e.g. the traffic is scheduled following the configured scheduling policy (e.g. PF) and end-to-end KPIs will be usefull to determine the whole state of the network, for instance if there are bottle necks leading to service-level agreement (SLA) violations. To achieve these applications a forecast of the traffic pattern and the users dispositions within the next short time window (e.g., next few minutes) is needed, a part from the network topology as we have seen. That is why monitoring data collected continuously from network devices and from service consumers/providers can be used for analysis purposes. In particular, ML models can be feed with these data to forecast relevant variables, e.g., the number of active users of each service and the position of the devices; since the SDN controller has the full view of a group of cells through their CSGw, mobility among cells can be predicted.

4.3 Application of the KPIs

The main purpose of the KPIs is to measure the quality of a user (end-to-end connection) is experiencing, making the possibility to simulate complex network scenarios that combines fixed and mobile networks. Thanks to these metrics the state of the network can be measured and decisions can be taken by a superior entity (multi-agent data analysis or MDA controller). To do that, the first step in the wireless CURSA-SQ wireless methodology, is to compute the number of aggregated users in a cell using each type of the considered traffics and which traffic types are sending/receiving each user/device. In order to achieve this characterization, monitoring data, service characteristics and users data are needed. After the disaggregated traffic estimation, the traffic is projected onto a future time window (forecasted) having the traffic generated by each user and their locations, which is used to propose an optimal entities configuration. Finally, with all the previous CURSA-SQ inputs estimated and parameters configured a simulation can be launched obtaining the end-to-end KPIs from the results. The entire process is summarized in Figure 7, which has in addition a refinement loop which tunes the internal and/or external parameters of the CURSA-SQ model in order to have a more accurate model. All these modules are explained in the following chapter.

4.4 Conclusions

Partial KPIs and end-to-end KPIs have been defined in terms of the logistic queue model of the previous chapter and the network topology introduced in this chapter. Thanks to them the network operation will be easier since the state of the network can be analyzed in different points of it. For instance, identifying a bottleneck in one of the access/metro nodes or the saturation of one of the cells connected to a CSGw.

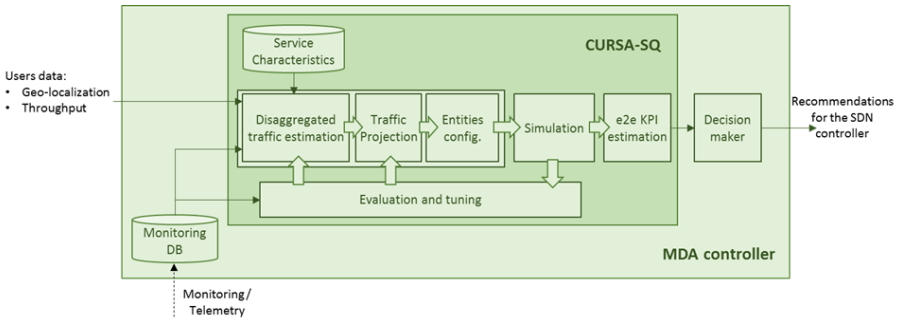

Figure 7: CURSA-SQ model for Wireless Scenarios.

We also have seen an application of the KPIs by running CURSA-SQ on a MDA controller which provides local information from the partial KPIs of each node and global information from the end-to-end KPIs to the SDN controller. The next step will be to define all the models and algorithms needed to run the CURSA-SQ simulation in order to obtain the aforementioned KPIs in mobile-fixed networks.

5. Algorithms, methods and models

5.1 Introduction

As we saw an extension of the logistic queue model is needed in order to use it in mobile networks. In these mobile networks or shared medium scenarios the different devices that are in a cell share the total capacity given by medium limitations. This means that the overall throughput traffic $Y(t)$ injected by the devices cannot exceed the medium capacity implying that every user will perceive a server bitrate that may change along time and it will not be constant anymore as it was model in the original CURSA-SQ methodology, chapter 3. Therefore, the extension needed to adapt CURSA-SQ for mobile scenarios will be detailed in the present chapter.

5.2 RAN model

The RAN can be model as a set of services, $\mathcal{S}$, and a set of regions or zones, $\mathcal{Z}$, where each region will have $|\mathcal{S}|$ generators injecting traffic of type $s \in \mathcal{S}$ to the RAN network. Similarly, each generator will be the aggregation of the users in region $z \in \mathcal{Z}$ and traffic type $s \in \mathcal{S}$. This aggregation of devices with the same service and not very separated can be made without any loss of accuracy because the users with traffic type s will experience the same behaviour if they are close enough because the rate given to a user will mainly depend on the position. So, the only approximation there will be is considering a limited set of zones but leading to a fine enough discretization, $\mathcal{Z}$, instead of considering the infinite possible dispositions in a cell. Thus, a cell topology, devices location and real or synthetic service traffic characteristics are needed in order to initialize the CURSA-SQ simulator with adaptations to wireless scenarios and obtain the already defined KPIs.

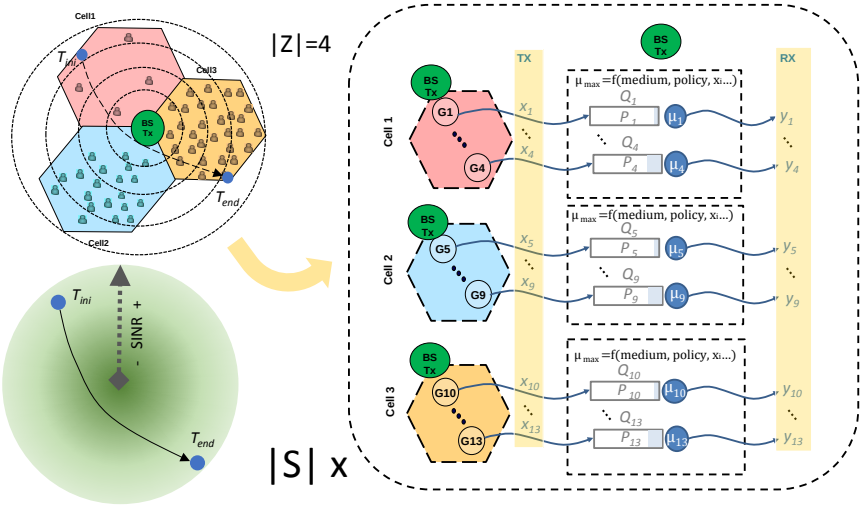

Figure 8: Wireless scenario for three sub-cells that share the same antenna, description of the problem.

Let us define an entity by the region $z \in \mathcal{Z}$ where it is, the service $s \in \mathcal{S}$ which belongs to and the number of users/devices $u(t) = \sum_i u_i(t)$ belonging to that entity. Recall that, according to [1], traffic generation can be fairly aggregated in entities with similar characteristics, including packet/flow traffic-

related, service-related, and infrastructure-related. Under that assumption, a shared medium of capacity C can be modelled as a system of queues, where we define a different queue for each group of devices consuming the same service. Then, given a set $\mathcal{S}$ of services and a set $\mathcal{Z}$ of groups of devices (zones), the cell can be modelled as a system of $N = |\mathcal{S}| \times |\mathcal{Z}|$ entities forming a topology. From the entities topology, N queues ruled by the logistic queue model with a μ that now mainly depends on the scheduling policy and the medium limitations is obtained, see Figure 8.

As a first consequence of this variation of the original problem, the actual server bitrate (μ_i) will not be constant because now we also have to accomplish the medium capacity. In fact, the actual server bitrate will change in those periods where $X(t) = \sum_i X_i(t)$ exceeds μ_{max} or if the scheduling policy requires, to introduce fairness for instance. Thus, if there is an excess of the medium capacity, the subsequent differential equations will not be time-invariant. So if we choose to model this scenario with a time-dependent μ we have to check that the theoretical properties of the CURSA-SQ model [10] are conserved.

5.3 Shared Capacity in Wireless Scenarios

The capacity of the cell is typically shared between the users taking into account proportionality and fairness. This means that the rate sharing will be proportional among all the devices which have backlogged traffic and fair because will penalize the entities with no backlogged traffic. In addition, every device perceives particular channel conditions from which a minimum server bitrate for every device can be ensured. The proportionality-fairness nature is brought to the logistic queue model allowing lagging flows to make up their lag by causing leading flows to give up their lead. Thus, the scheduled rate to a given entity i will be

$$\mu_i(t) = C \cdot \left[(1-\alpha) \cdot f_i + \alpha \cdot g\left(Q_i^0(t), Q^0(t)\right)\right] \tag{13}$$

where $\alpha \in [0,1]$ is a factor which weights how much impact has the variable rate into the final scheduled rate to entity i, f_i is the fixed proportion of the capacity of the cell that entity i will perceive, which depends on the distance between the group and the antenna, among other factors impacting the SINR, and $g\left(Q_i(t), Q(t)\right)$ is the policy function that models the cell's scheduler with $Q_i^0(t)$ being the state of the local queue and $Q^0(t)$ being that of all queues in the cell if the server bitrate were distributed following no policy at all (attending only to physical factors such as SINR or distance). For the particular case studied of a PF-policy this function will correspond to equation 18.

Therefore, the rate will be scheduled in two steps: 1) distribute the total capacity following a non-fairness criteria e.g. applying equation 13 with $\alpha = 0$ and run CURSA-SQ with fixed time server rates, 2) taking the state of the queues from the simulation in step 1 and distributing the server bitrate with the proportional fair policy from equation 13 when $\alpha \neq 0$.

Proposition 5.1. Fairness factor. *Under the assumption of using a PF-policy for the traffic scheduling the α in equation 13 can be seen as a fairness factor. Moreover, it can be computed taking into account the states of the queues in stage 1 as,*

$$\alpha(t) = \frac{1}{|\mathcal{Z}| \cdot |\mathcal{S}|} \cdot \sum_i \frac{Q_i^0(t)}{k_i}. \tag{14}$$

Proof. Let us assume we have $|\mathcal{Z}| \cdot |\mathcal{S}|$ entities whose server bitrate is given by the scheduler following a PF-policy. Then, the higher the per-entities queue load is, computed as $Q_i^0(t)/k_i \in [0,1]$, the more balance should be provided among the queues, which is provided by equation 18.

On one hand, looking at the limits of the entity load function for all the entities, $Q_i^0(t) \longrightarrow 0$ (i.e. $\alpha(t) \longrightarrow 0$) there would not be any backlogged traffic to balance and introduce fairness to the system and the initial rate distribution of step 1) is sufficient to provide a fair distribution. On the other hand, $Q_i^0(t) \longrightarrow k_i$ (i.e. $\alpha(t) \longrightarrow 1$) means that almost every entity starts to lose traffic or have lost traffic needing a load balance between the queues. For an intermediate case, if at least one queue has queued traffic then $\alpha(t) \neq 0$ and a balance with respect to the empty queues is done. □

In equation 13 the first term of the sum, f_i, models the fixed rate given to an entity in terms of the channel state perceived by the entity. That is why a first assumption must be made, that is each flow has perfect knowledge on channel state because otherwise this minimum rate could not be computed. Under this assumption the scheduler will know the SINR perceived by each entity and compute the proportion of fixed rate from the total capacity corresponds to each entity, being

$$f_i = \frac{SINR_i}{\sum_i SINR_i}. \tag{15}$$

Since the *SINR* function is a bounded function, f_i will be a bounded function for general cases. In addition, for some particular cases such as the FRIIS radiation model, the function will be monotonically decreasing. Note that, if $\alpha = 0$ all the scheduled rate will correspond to the proportion given by each channel condition perceived by the entities; in other words, $\mu_i(t) = C \cdot f_i$.

The second term of the sum, $g\left(Q_i(t), Q(t)\right)$, of equation 13 combines the proportionality among all those entities with queue state, $Q_i(t)$, different from zero and fairness because the entities with queue state, $Q_i(t)$, equal to zero will not be given variable rate, having only the minimum fixed rate from the first addend. Note that, if $\alpha = 1$ all the cell capacity is scheduled following the PF policy without taking in to account channel conditions implying scheduled rate equal to zero for some entities at certain instants of time.

To understand this second part of the model, let us begin with the proportional scheduling which corresponds to assume an initial given distribution of fixed rate (with no dependence with respect to time). Then, computing the output traffic of each entity, $Y_i(t)$, the entities server rate is recalculated in terms of the initial distributions, the output traffic and the total capacity which is shared between entities.

Proposition 5.2. Proportional scheduling *Let us assume we have $|\mathcal{Z}| \cdot |\mathcal{S}|$ entities with its corresponding traffic generators, $X_i(t)$, one stage of queues with $|\mathcal{Z}| \cdot |\mathcal{S}|$ queues, every queue with its maximum capacity k_i, current state $Q_i(t)$ and server rate μ_i, and a maximum medium capacity μ_{max} proportionally shared. Then the proportionally distributed server bitrate will be,*

$$\begin{aligned} Y_i^0(t) &= min\left\{\hat{X}_i(t), \mu_i^0\right\} \\ \mu_i(t) &= min\left\{\mu_i^0, \mu_{max} \cdot \frac{Y_i^0(t)}{\sum_i Y_i^0(t)}\right\} \end{aligned} \tag{16}$$

where $\hat{X}_i(t)$ is the input traffic of the finite queue (for more details see equation 4) and $Y_i^0(t)$ is the maximum output bitrate of the i-th queue under finite server bitrate and no shared medium limitations.

Proof. This result can be easily proven by taking into account the output bitrate limitations. On one hand, we have that the maximum output bitrate given by the server is μ_i. On the other hand, we have that the maximum output bitrate given by the shared medium is $\mu_{max} \cdot \frac{\hat{Y}_i(t)}{\sum_i \hat{Y}_i(t)}$ whenever there is excess ($\sum_i \hat{Y}_i(t) > \mu_{max}$) and μ_i otherwise. Combining both limitations we reach to the equation 16. Note that,

the shared medium limitation add to the actual output bitrate, $\hat{\mu}_e$, a dependency with respect to time making the ODE equation 5 non-autonomous. □

Remark that proposition 5.2 corresponds to a proportional sharing between all the entities which does not implies a fair distribution of the total capacity. In fact, it is easy to prove that this method is not fair because the more input an entity has the more server bitrate will be given disregarding the rest of the entities which will be penalized for not having so much input. The fairness condition is imposed reducing the variance between the backlogged traffic of devices. In fact, a fairness index can be defined in terms of the throughput of the queue.

Definition 5.3. Jain's Index gives an indication of the variation in throughput between users for similar per-user input traffic. For a given vector $X \in \mathbb{R}_+^N$ the Jain's fairness index $J : \mathbb{R}_+^N \to \mathbb{R}_+$ is given by

$$J(Y) = \frac{\left(\sum_{i=1}^{N} Y_i\right)^2}{N \sum_{i=1}^{N} Y_i^2} = \frac{\bar{\mathbf{Y}}^2}{\bar{\mathbf{Y}}^2 + var(\mathbf{Y})} \in \left[\frac{1}{N}, 1\right], \tag{17}$$

where $\bar{\mathbf{Y}}$ and $var(\mathbf{Y})$ are the mean and the variance respectively of the average user throughputs.

Note that, the more similar the average user throughput (i.e. $var(\mathbf{Y}) \longrightarrow 0$), the higher the value of the Jain's Index will be. $J = \frac{1}{N}$ corresponds to the least fairness where only one user is benefited from the scheduling and $J = 1$ corresponds to the most fairness case where all the users are benefited from the rate distribution.

Proposition 5.4. *Under the assumption of having N entities injecting traffic with the same characteristics, $var(\mathbf{X}) \sim 0$, and a non-uniform per-entity rate distribution (e.g. the SINR based of equation 15) the minimization of the variance of the throughput, $var(\mathbf{Y})$, leads to the minimization of the variance of the queued traffic, $var(\mathbf{Q})$. Note that, $\mathbf{X} \in \mathbb{R}^N$ corresponds to the input traffics of all the entities for a given instant of time t.*

Proof. Three cases must be consider: no entity has backlogged traffic, all the entities has backlogged traffic and some entities has backlogged traffic. By noting that the queue state is a bounded function between 0 and k the queue capacity, in the first case the variance of the backlogged traffic will correspond to the lower bound, $var(\mathbf{Q_1}) = 0$. For the last two cases the variance of the third case will be greater or equal to the variance of the second case, $var(\mathbf{Q_3}) \geq var(\mathbf{Q_2})$, since in the third case $Q_e(t) \in (0, k]$ whereas in the second case $Q_e(t) \in [0, k]$. Having, on one hand, that $var(\mathbf{Q_1}) \leq var(\mathbf{Q_2}) \leq var(\mathbf{Q_3})$ so we need to prove that $var(\mathbf{Y_1}) \leq var(\mathbf{Y_2}) \leq var(\mathbf{Y_3})$ and the implication will be proven.

For the case where no entities have backlogged traffic, the throughput for all the entities will be $Y_e(t) = X_e(t)$ so $var(Y) = var(X) \sim 0$. For the second and the third cases is better to work with $var(\mathbf{Y}) + \bar{\mathbf{Y}}^2 = N \sum_{i=1}^{N} Y_i^2$ and proof that $N \sum_{i=1}^{N} Y_{2,i}^2 \geq N \sum_{i=1}^{N} Y_{3,i}^2$, corresponding to the second and third case respectively, is straightforward since in the second case the output will always be $Y_i(t) = \mu_i$ for all the entities whereas in the third case the summation can be split into those entities with backlogged traffic with output traffic corresponding to $Y_i(t) = \mu_i$ for $i \in \mathcal{B}$ and those entities with empty queue with output traffic $Y_i(t) = X_i(t) < \mu_i$ for $i \notin \mathcal{B}$. □

The proposition 5.4 can be generalized for input traffic with different characteristics by taking into account the variance of the input traffic, $var(\mathbf{X})$. Therefore, in order to reduce the variance of the queued traffic two sets can be considered, the set of queues with backlogged traffic, $\mathcal{B}$, and the set of queues with no backlogged traffic, $\bar{\mathcal{B}}$.

Proposition 5.5. Proportionally Fair Scheduling *Let us assume we have N entities with its corresponding traffic generators, $X_i(t)$, one stage of queues with N queues, every queue with its maximum capacity k_i, current state of the backlogged traffic in the queues, $Q_i^0(t)$, under a proportional distribution of the rates μ_i^0, and a maximum medium capacity μ_{max} fairly shared. Then the Fluid Flow Fair Queuing (FFQ) model for the server bitrate is,*

$$\mu_i^{FFQ}(t) \equiv g\left(Q_e(t), Q(t)\right) = \frac{w_i \cdot \Gamma(Q_(t))}{\sum_j w_j \cdot \Gamma(Q_j(t))} \tag{18}$$

where C is the total capacity shared between the entities (queues), w_i are the scheduling weights and $\Gamma(x)$ is the step function defined as 0 for $x \in (-\infty, 0]$ and 1 for $x \in (0, +\infty)$. Note that, the step function, $\Gamma(x)$, can be defined as a smooth, continuous and integrable function as has been done in the logistic queue model for equation 4.

Proof. Since the proportionally fair scheduling is defined as a proportional rate for those flows with backlogged traffic and no rate for those flows with empty queue there is only needed to see that equation 18 satisfies two properties: that whenever a flow has no backlogged traffic its rate will be zero and the following,

$$\left| \frac{\mu_i^{FFQ}(t)}{w_i} - \frac{\mu_j^{FFQ}(t)}{w_j} \right| = 0, \quad \forall i, j \in \mathcal{B}.$$

Both properties can be proven by taking into account the definition of the step function; for instance, if the queue of an entity i is zero $\forall t$ then the rate given to that entity, $\mu_i^{FFQ}(t)$, whereas an entity with backlogged traffic will perceive a rate $\mu_i^{FFQ}(t) = \frac{w_i}{\sum_{j \in \mathcal{B}} w_j}$ which satisfy the second property. □

The idealization becomes because each flow has perfect knowledge on channel state, the channel is also an error-free channel and there is a perfect MAC protocol. Remark that, from equation 18, if one entity has backlogged traffic will share the total capacity C of the scheduling area (normally a cell) between those queues which has traffic in their queues taking into account their respective importance (weights w_i); otherwise, the rest of the queues which do not have backlogged traffic will not be given capacity by the scheduler.

Example 5.1. Let us assume we have N entities, a PF scheduler, i.e. the server rate given to each entity is ruled by equation 13, and the users perceiving the same SINR, i.e. $f_i = 1/N \; \forall i$. Then, taking into account that $g_i \in [0, 1]$ depending on the queue state of the entity and the rest of entities, we have that:

$$\frac{1-\alpha}{N} \cdot C \leq \mu_i(t) \leq \frac{1+\alpha(N-1)}{N} \cdot C.$$

Note that, the higher the fairness factor is the wider the rate bounds will be whereas for low fairness factors the server rate will present narrower bounds.

Example 5.2. Let us assume we have N entities, a PF scheduler, i.e. the server rate given to each entity is ruled by equation 13, and a fairness factor $\alpha = 1$. Then, taking into account the maximum and the minimum of the weights of the policy function, $lb = min(\mathbf{w})$ and $ub = max(\mathbf{w})$, the policy function can be bounded as,

$$\frac{lb}{ub \cdot |\mathcal{B}|} \leq g\left(Q_e(t), Q(t)\right) \leq \frac{ub}{lb \cdot |\mathcal{B}|}.$$

So, if the weights are the same for all the entities and all the entities have backlogged traffic, i.e. $\forall i \in \mathcal{B}$ and $|\mathcal{B}| = N$, then the rate will be distributed uniformly among all the entities, in other words

$$g\left(Q_e(t), Q(t)\right) = \frac{1}{N}.$$

Example 5.2. Consider an extension of the previous examples where now the weights are unknown and the users do not have to be in the same location. Therefore, the user with the minimum SINR ($f_i = 0$) will provide the lower bound and the user with the highest SINR ($f_i = 1$) will provide the upper bound. Now, defining the rate given by the SINR perceived by the user as $\mu^{SINR} = C \cdot (1 - \alpha) \cdot f_i$ and the rate given by the PF policy function as $\mu^{PF} = C \cdot \alpha \cdot g\left(Q_e(t), Q(t)\right)$ we can obtain the bounds of the scheduled rate for an entity i. The first addend of equation 13 corresponds with,

$$0 \leq \mu^{SINR} \leq (1 - \alpha) \cdot C,$$

whereas the second addend of equation 13 corresponds with,

$$\alpha \cdot \frac{C}{N} \cdot \frac{lb}{ub} \leq \mu^{PF} \leq \alpha \cdot C \cdot \frac{ub}{lb},$$

which results into a total scheduled rate bounded by,

$$\begin{cases} \alpha \cdot \dfrac{C}{N} \cdot \dfrac{l}{ub} \leq \mu_i \leq \left(1 + \alpha \cdot \dfrac{ub - lb}{lb}\right) \cdot C \ \text{ for } \ i \in \mathcal{B} \\ 0 \leq \mu_i \leq \alpha \cdot C \ \text{ for } \ i \notin \mathcal{B} \end{cases}$$

Once the scheduled rate is bounded we can compute the bounds of the KPIs. The bounds for the throughput and the loss are easy to compute since they correspond to loose nothing or loose everything, i.e. $0 \leq L_{RAN}(t) \leq 1$. For the delay, the bounds will be expressed in terms of the bounds of the scheduled rate as

$$0 \leq D_i(t) = 8 \cdot \frac{Q_i(t}{\mu_i(t)} \leq \frac{8 \cdot k_i(t)}{\mu_i(t)},$$

where $k_i(t) = u_i(t) \cdot k_s$ is the queue capacity of entity i with $u_i(t)$ its number of users and k_s the characteristic queue capacity for service s which entity i belongs to. Hence,

$$max(D_i(t)) = \frac{8 \cdot k_i(t)}{min(\mu_i(t))},$$

with $min(\mu_i(t))$ one of the previously computed in the examples.

5.4 Disaggregated Traffic Estimation

Let us assume we have a set of zones, $\mathcal{Z}$, and a set of services, $\mathcal{S}$. Then, to make possible the aforementioned aggregation of devices into groups of devices that share similar location and the same service a disaggregated traffic estimation is needed. The disaggregation of traffic consists on solving an optimization problem for every cell c, that can be stated as follows.

Given:

- the measured throughput of the cell c defined as a set of statistics (W_c), including mean, max, min, variance, 95^{th} percentile, etc.,
- the total number of users in a cell and an initial discretization Z_c of the cell in terms of the perceived SINR,
- a set of services S and their traffic characteristics,

- an initial percentage (v_{cs}) of traffic volume for every service s. This initial percentage can be used to consider smooth evolutionary solution when coming from the previous time interval, as well as serve as a way to bias the optimization problem.

Output: the percentage of traffic volume for every pair $< z, s >$ for the cell c (θ_{csz}).

Objective: minimize the weighted error between the given throughput and the estimated one in terms of the given statistics, as well as the weighted error between the initial per-service traffic percentages and the estimated ones.

The objective function Φ_c is defined for every cell c:

$$\Phi_c = \gamma_1 \cdot \sum_{w \in W_c} (f_w(\theta, Z_c, S) - W_c(w))^2 + \gamma_2 \cdot \sum_{s \in S} \left(v_{cs} - \sum_{z \in Z} \theta_{csz} \right)^2, \quad \forall c \in C \tag{19}$$

where, specific functions $f_w(\cdot)$ are defined to compute the average for every traffic statistic given the unknown percentages of traffic volume, the initial discretization, and the characteristics of the services. The disaggregated traffic estimation optimization problem can be solved using the gradient descent algorithm [11]. The estimation results, together with the set of statistics for the measured throughput, are stored in a database.

5.5 Traffic projection

The traffic projection module applies interpolation to the estimated historical disaggregated traffic and measured cell throughput to forecast per $< z, s >$ pair traffic and cell throughput in the next time window. The size of the historical data considered for interpolation is defined by parameter h_w.

This module uses structural models based on the state space models, which allow using more than one correlated time series [12]. In particular, the accuracy of interpolations of per $< z, s >$ pair estimated traffic increases when the measured cell throughput is considered, since the aggregation of all pairs $< z, s >$ is naturally correlated to the overall cell throughput. The structural model of a time series y_t with frequency f will be as follows, where μ_t is the trend, β_t is the slope and s_t is the seasonal component (only one since each entity corresponds to only one type of service).

$$\underbrace{\begin{pmatrix} \mu_t \\ \beta_t \\ s_t \\ s_{t-1} \\ \vdots \\ s_{t-f+2} \end{pmatrix}}_{x_t} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -1 & -1 & \dots & -1 & -1 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix}}_{\phi} \cdot \underbrace{\begin{pmatrix} \mu_{t-1} \\ \beta_{t-1} \\ s_{t-1} \\ s_{t-2} \\ \vdots \\ s_{t-f+1} \end{pmatrix}}_{x_{t-1}} + \underbrace{\begin{pmatrix} w_t^{(1)} \\ w_t^{(1)} \\ w_t^{(1)} \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{w_t},$$

for the transition state equation, where $\hat{w}_t \sim N\left(0, \begin{pmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{pmatrix}\right)$ corresponds to the first three components of the w_t noise vector, and

$$y_t = \underbrace{\begin{pmatrix} 1 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}}_{A_t} \cdot x_t + v_t,$$

for the observation equation, where $v_t \sim N(0, R)$. Note that all these parameters $\Theta = (q_{11}, q_{22}, q_{33}, R)$ should be estimated by maximizing a likelihood function.

Moreover this structural model can be combined with a proxy series, such as the total cell throughput, to increase the model accuracy. Thus, the model would be a combination of the upper model with the following.

$$x_t = \begin{pmatrix} x_t^{(1)} \\ x_t^{(2)} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix} \cdot \begin{pmatrix} x_{t-1}^{(1)} \\ x_{t-1}^{(2)} \end{pmatrix} + \begin{pmatrix} w_t^{(1)} \\ w_t^{(2)} \end{pmatrix},$$

for the transition equation for the state, where $w_t = \begin{pmatrix} w_t^{(1)} \\ w_t^{(2)} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix} \right)$, and

$$y_t = \begin{pmatrix} y_t^{(1)} \\ y_t^{(2)} \end{pmatrix} = A_t \cdot x_t + v_t,$$

for the observation equation, where $v_t \sim N(\mathbf{0}_{2\times 1}, \mathbf{Q}_{2\times 2})$ and A is equal to the identity when all the data is observed (there is no missing, a missing treatment could be added). Note that, all these parameters $\Theta = (\phi_{11}, \phi_{12}, \phi_{21}, \phi_{22}, q_{11}, q_{12}, q_{21}, q_{22})$ should be estimated by maximizing a likelihood function.

In addition, the method can potentially identify the structural components of the considered time series (e.g. trend or seasonality) making possible to deal with very different behaviors with just one single modelling approach.

Note that this part is optional depending on the application of the model. If the model is used to take real time operation decisions on to the network a traffic projection is needed. Otherwise, if the application is only to study the behaviour of the network given an scenario no traffic projection will be needed.

5.6 Entities Configuration

By fixing a finite number of zones to consider in a cell the entities topology can be computed from the cell topology (which includes the SINR map, physical characteristics of the cell, etc) and the devices location, as it is shown in Figure 8. Recall that, an entity was defined as a generator (defined by the type of service, the region where it is and the number of users it has), a queue (with its queue capacity) and a server bitrate given by the scheduler and/or policy (e.g. priorities) considered. Then, the problem is reduced to $N = |\mathcal{S}| \times |\mathcal{Z}|$ entities each one with a logistic queue model. Note that, each entity injects a traffic which is proportional to the number of users that are in that region for an specific instant of time. In fact, the traffic injected by an entity can be modelled as,

$$X_e(t) = U_e(t) \cdot X_s(t; TC_s) \tag{20}$$

where $U_e(t)$ are the number of users entity e has along time and TC_s are the traffic characteristics of service s. For instance, in Figure 8, $12 \times |\mathcal{S}|$ entities are considered because the number of zones is fixed to 4 and the antenna provides service to 3 sub-cells (or sectors). The initial discretization of the cell, Z_c, is very fine implying a high number of entities and a high computational cost. Due to that reason a reduce in the number of possible generators to configure for the simulation step by grouping $< z, s >$ pairs with similar characteristics to create mobile entities is necessary. This problem can be stated as follows.

Given:

- the set of pairs $< z, s >$ and the projection of traffic for every pair,
- a number $|N_c|$ of entities to be created by grouping $< z, s >$ pairs with similar perceived SINR.

Output: the $|N_c|$ (multiple of $|\mathcal{S}|$) mobile entities, including its assigned SINR and its projected traffic.

Objective: minimize the error between the SINR assigned to each entity and the SINR of each $< z, s >$ pair weighted by the number of the pair. As a secondary objective, we are interested in obtaining entities representing a balanced number of users.

The entities configuration optimization problem can be solved using the k-means clustering algorithm [13] complemented with a final balancing phase focused on the secondary optimization objective. For more detail, let us consider a particular model where the set of pairs $< z, s >$ is given by the mapping $\mathcal{U} \rightarrow \mathcal{Z} \cup \mathcal{S}$, with $\mathcal{U}$ the set of users/devices, and the time $t \in [T_{ini}, T_{end}]$ is taken into account.

Then, the entities must be placed in those locations where the expected SINR (computed from the SINR map of the cell and the users' dispositions) and the assigned is minimized in order to achieve the least possible error. Assuming that from user devices location forecasts and cell/sectors configuration, the expected SINR π of every device at every time instant can be computed. It is also assumed, for the sake of simplicity, that every device injects/receive traffic for one and only one service during the period T.

Under these assumption a k-means can be proposed to seek the optimal entities topology with the following sets,

- $\mathcal{U}$: set of users
- $\mathcal{S}$: set of services
- $\mathcal{E}$: set of entities
- T: time interval

the following parameters,

- δ_{is}: 1, if user/device $i \in \mathcal{U}$ belongs to service $s \in \mathcal{S}$,
- π_{it}: expected SINR of user/device $i \in \mathcal{U}$ in time $t \in T$,

and the following variables,

- x_{ijt}: 1, if the user/device $i \in \mathcal{U}$ belongs to entity $j \in \mathcal{E}$ at a certain instant of time $t \in T$,
- y_{js}: 1, if entity $j \in \mathcal{E}$ belongs to service $s \in \mathcal{S}$,
- z_j: SINR of the entity $j \in \mathcal{E}$.

Then the optimal entities topology, in terms of the SINR, will be the one which

$$\min \sum_{i \in \mathcal{U}} \sum_{t \in \mathcal{T}} v_{it} \tag{21a}$$

$$\text{subject to:} \quad \sum_{j \in \mathcal{E}} x_{ijt} = 1 \ \ \forall i \in \mathcal{U}, t \in T \tag{21b}$$

$$x_{ijt} \leq \sum_{s \in \mathcal{S}} \delta_{ij} y_{js} \; \forall i \in \mathcal{U}, j \in \mathcal{E}, t \in T, \tag{21c}$$

$$\sum_{s \in \mathcal{S}} y_{js} \leq 1 \forall j \in \mathcal{E}, \tag{21d}$$

$$v_{it} \geq \pi_{it} - z_j - M \cdot (1 - x_{ijt}) \forall i \in \mathcal{U}, j \in \mathcal{E}, t \in T, \tag{21e}$$

$$v_{it} \leq -(\pi_{it} - z_j) - M \cdot (1 - x_{ijt}) \forall i \in \mathcal{U}, j \in \mathcal{E}, t \in T, \tag{21f}$$

where the first constraint 21b ensures that every user device is assigned to one and only one entity at every time, the second constraint 21c guarantees that devices are assigned to entities belonging to the sane service, the third constraint 21d establish that entities must belong to one single service and the last two constraints 21e and 21f link the previous constraints with the minimization variable v_{it} by computing the absolute value of the difference between the expected and the assigned SINR. Note that, M is a value big enough to invalidate the inequalities when the the device does not belong to the entity. The output of this k-means problem will be the entities topology and the number of users in every entity.

Finally, all the parameters needed to configure the entities for a CURSA-SQ simulation can be obtained from a solution of the problem. For instance, the queue capacity must be scaled taking into account the number of users a generator has at every instant of time being $k_{e_j} = k_{s_j} \cdot u_j(t)$ with k_{s_j} the queue capacity associated to a given service. Note that, VoD will not have the same buffer allocation than instant messaging and that is why queue capacity is associated to the service and scaled by the number of users that are consuming that service.

5.7 Priority Queues

A part from the scheduling algorithm in the RAN networks, interfaces can be dedicated to more than one service, each service with different priorities. So, the capacity of that interface (μ_{max}) must be shared between all the services, serving first the queue with the highest priority and last the queue with the lowest priority. These priorities will lead us to have time dependent server bitrates.

Let us assume we have $i = 1, \cdots, N$ queues each one with its current state, $Q_i(t)$, initial state, $Q_i(t_0)$, its individual server bitrate μ_i^0 and its priority, p_i. So we proceed by computing each queue and its throughput using the models previously defined, but first we have to compute the actual value of each server bitrate due to priority constraint, $\mu_i(t)$. Due to the sequential nature of this scenario, each server bitrate will depend on the previous queue if we sort the queues in descendant order of priority, having:

$$\begin{aligned} &\mathbf{I} = \text{sort}(\mathbf{P}), \;\; \text{with} \;\; \mathbf{P} \;\; \text{the priorities of each queue} \\ &C_{I(1)} = \mu_{max} \longrightarrow \mu_{I(1)} = min\left\{\mu^0_{I(1)}, C_{I(1)}\right\}, \text{and for} \;\; i \geq 2 \\ &C_{I(i)}(t) = C_{I(i-1)} - Y_{I(i-1)}(t) \longrightarrow \mu_{I(i)}(t) = min\left\{\mu^0_{I(i)}, C_{I(i)}(t)\right\}, \end{aligned} \tag{22}$$

where $C_{I(i)}$ is the corresponding interface capacity after being served the queues with a higher priority than queue $I(i)$. It can also be explained as the corresponding server bitrate to queue $I(i)$ just taking into account the priorities restriction and the maximum interface capacity μ_{max}.

As we need the outflow from the previous less priority queue we will have to run CURSA-SQ sequentially following the order in $\mathbf{I}$ with the server bitrate μ_i, $i \in \mathbf{I}$, previously defined. Therefore, we just have to distribute sequentially the server bitrates between all the queues which share the same interface attending the priorities and its individual server bitrates, as can be seen in equation 22.

Proposition 5.6. Non-priority constraint. *If $\mu_{max} \geq \sum_j \mu_j^0$ then there will be no priority restriction because the initial values for each server bitrate are more restrictive than the priority constraint, i.e. $C_j \geq \mu_j^0 \ \forall j \in \mathbf{I}$.*

Proof. For $i = 1$, it can be easily proven since $C_1 = \mu_{max} \geq \sum_j \mu_j^0 \longrightarrow C_1 \geq \mu_1^0 = \mu_1$. For $i \geq 2$, can be proven by applying the output bitrate bounds $0 \leq Y_j(t) \leq \mu_j(t)$ to the third equation in 22 and obtaining

$$C_j(t) + \mu_{j-1}(t) \geq C_{j-1}(t) \longrightarrow \sum_{j=2}^{k} \mu_{j-1} \geq \sum_{j=2}^{k} [C_{j-1} - C_j],$$
$$\sum_{j=2}^{k} \mu_{j-1} \geq C_1 - C_k, \quad C_k \geq C_1 + \sum_{j=2}^{k} \mu_{j-1}$$

and making use of $C_1 = \mu_{max} \geq \sum_{j=1}^{N} \mu_j$ and using the non-negativity of the server bitrate, $\mu_i \geq 0$, we obtain

$$C_k \geq \sum_{j=1}^{N} \mu_j + \sum_{j=2}^{k} \mu_{j-1} = \sum_{j=k}^{N} \mu_j \geq \mu_k$$

and since $\mu_j \geq \mu_j^0 \ \forall j \in \mathbf{I}$ the proposition is proven. □

Proposition 5.7. Priority constraint. *If $\mu_{max} < \sum_{j=1}^{N} \mu_j^0$ then there will exists at least one queue such that its actual server bitrate will be zero due to the priority constraint, in other words $\exists k \in \mathbf{I}$ s.t. $\mu_j = 0$, $\forall j \geq k$. Which implies that $C_j(t) = 0$ for those $j \geq k$, i.e. $\mu_j^0 \geq C_j(t)$ for those $j \geq k$.*

Proof. Similarly to the proof done for proposition 5.6 it can be established that,

$$C_j(t) + \mu_{j-1}(t) \geq C_{j-1}(t) \longrightarrow \sum_{j=2}^{k} \mu_{j-1} \geq C_1 - C_k$$
$$C_{j-1}(t) \geq C_j(t) \longrightarrow C_k \geq C_N \geq \mu_{max} - \sum_{j=1}^{N} \mu_j(t)$$

which, considering $C_1 = \mu_{max}$ and the non-negativity of the server bitrates (i.e. $\mu_j(t) \geq 0$), leads to

$$\sum_{j=2}^{k} \mu_{j-1}(t) \geq \sum_{j=1}^{N} \mu_j(t)$$

$$0 \geq \sum_{j=1}^{N} \mu_j(t) - \sum_{j=2}^{k} \mu_{j-1}(t) = \sum_{j=k}^{N} \mu_j(t) \geq 0$$

$$\mu_j = 0 \ \ \forall j \geq k$$

□

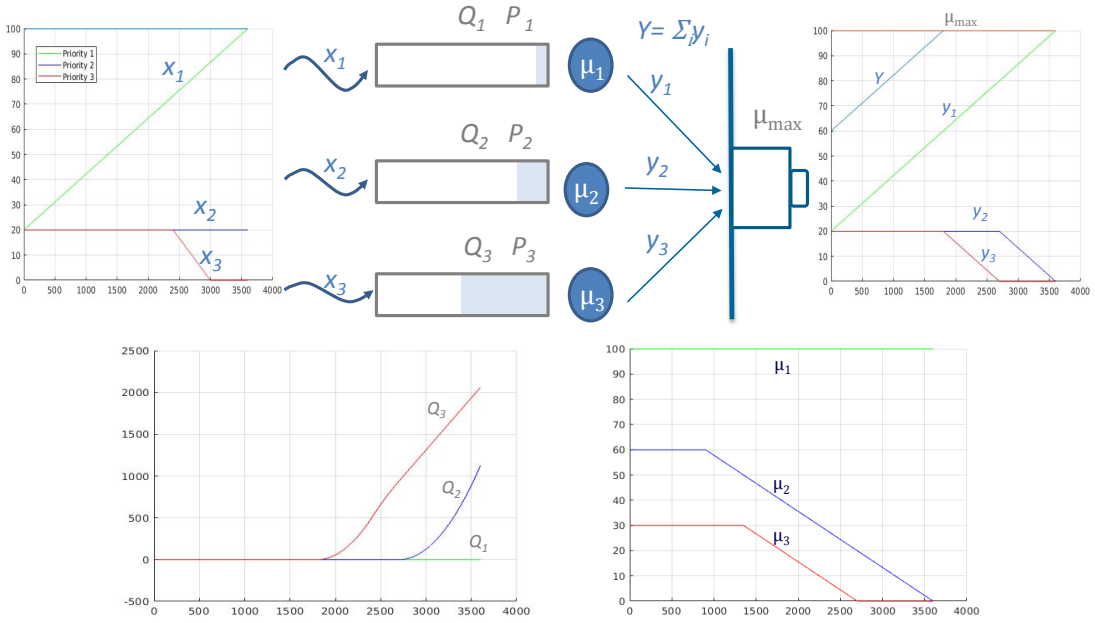

Figure 9: Priority queues scenario.

As a result we have that if we consider 3 queues with different priorities each one, **P**, initial queue value, $\mathbf{Q_0}$, set to zero and a maximum interface speed of $\mu_{max} < \sum_i \mu_i^0$ we will obtain more backlogged traffic in the less important queues, see Figure 9. We also can see the implications of proposition 5.7, which tell us that at least one queue will be set to have $\mu_j = 0$, in this case $j = 3$ the one with the less priority. In fact, we can see the scenario as the maximum interface speed, μ_{max}, being scheduled giving to the 1st one all the possible speed, then to the second the speed which have left and so on. Then, if there is a generator injecting traffic to a queue with a very low priority and the interface is saturated, the traffic might have to wait until the interface becomes less saturated (proposition 5.7). As a counterpart, the server-bitrate will be time-dependent, unless the first one, because is computed as the difference between the leftover capacity and the output traffic. This means that the integration process will be slower as it is solving a non-autonomous ODE, but it can be speed up by considering time-dependent server-rate only when there is not a fulfillment of the maximum interface capacity (priority constraint).

5.8 Evaluation and tuning

Once the entire model of the network has been explained let us imagine that we want to estimate the KPIs for a next time window. Then, these results can be temporally stored and used to evaluate their precision by comparing them against real traffic conditions measured from the network. Note that the dynamic configuration module requires different configuration parameters that need to be tuned according to the specific scenario under study. Specifically: i) the disaggregated traffic estimation submodule includes parameters γ_1, γ_2 and v_{cs}. Note that the value of parameter v_{cs} can be equal to the values of the θ_{csm} from the previous estimation window so the values selected for γ_1 and γ_2 can result into more dynamic or more persistent models; ii) in the traffic projection submodule, the parameter to evaluate and tune is the size of the historical time window, h_w; using large historical time windows, more importance will be given to the trend in the model whereas with small sizes the model will detect changes in the injected traffic to the network; and iii) the $|N_c|$ parameter can be adjusted in the entities configuration submodule; note that the higher the number of entities, the higher the accuracy of the simulated traffic characteristics.

Let us assume that some measurements of the traffic are available at the input of the access/metro network (traffic from the RAN)and some measurements of the aggregated traffic per service that can be obtained, e.g., at the output of the mobile core. Note that, other monitoring points can be considered as well but we consider these ones because the monitoring data they provide has a coarser granularity than that from the mobile network, which uses fine grained data for near real-time management.

Therefore, in order to evaluate the current value of the above configuration parameters, an optimization problem is solved to find the optimal value of the configuration parameters, $Params^*$, for the estimation window corresponding to the monitoring data. The optimization problem is based on comparing the traffic measured at different points during the CURSA-SQ simulation and KPI estimation against the traffic that is measured from the network. Then, the optimization problem for evaluation can be stated as follows:

Given:

- the measured traffic at the interconnection between the RAN and the access/metro network for a given time period T,
- the aggregated traffic per service monitored at the output of the access/metro network during the same time period T,
- the traffic measurements form the CURSA-SQ simulation module for time period T.

Output: the set of optimal CURSA-SQ configuration parameters, $Params^* =< \gamma_1^*, \gamma_2^*, v_{cs}^*, h_w^*, |N_c|^* >$.

Objective: minimize the distance between the characteristics of the traffic observed in the real network and those of the traffic generated during the simulation.

Once solved the evaluation problem, tuning of the configuration parameters can be performed for the next estimation window. As the optimal values for the configuration parameters correspond to a historical time period T, the tuning module stores the optimal values just obtained and predicts the values of the parameters to the next time window based on their historical evolution. Then, considering a confidence interval for such predictions, parameters are updated only if their current value is outside the corresponding confidence interval.

5.9 Integration Methods

If the FFQ server rate sharing model of equation 13 is plugged directly into the CURSA-SQ equations of Section 3.3, CURSA-SQ model will correspond with a coupled system of non-autonomous ODEs. For few entities the resulting system of few coupled ODEs might not be difficult to solve, but for a large number of entities to compute the solution will be hard.

Thus, in order to decouple the system assume different time intervals are taken, $\Delta T^k \in T = [T_{ini}, T_{end}]$ with $k = 1, ..., NS$ the number of time intervals. Then, two stages are considered: a first solution is obtain at a constant server bitrate and a second result is obtain applying equation 13 which introduces the scheduling policy. Depending on the length of the time intervals, ΔT^k, the scheduled rate will variate less, for long time periods, or more, for short time periods.

Therefore, in order to compute scheduled server bitrates is necessary to run CURSA-SQ with an initial time-invariant server bitrates and then compute the final solution with the time-variant server bitrates. The time-invariant bitrate will depend on the case study (uniform, depending on physical factors, ...) and the time-variant bitrate will depend on the time-invariant bitrate and the scheduler chosen (RR, PF ...).

Depending on the integration approach the system of equations can be transformed into an autonomous system of decoupled ODEs. Then, two main methods can be distinguished: the time-invariant approximation and the time-variant approach. From the time-invariant approximation we can take to paths: considering small slices or taking into account when there is excess or the sharing changes among the devices.

Data: $t \in [t_0, t_{max}], Q_{0,i}, \mu_i, \mu_{max}$
Result: $Y_i(t), Q_i(t)$
for *every queue, i,* **do**
  run CURSA-SQ with server bitrate μ_i;
end
for *every queue, i,* **do**
  computation of $\hat{\mu}_i(t)$ using equation 16;
  for *every period, k, of length ΔT* **do**
    computation of the $\hat{\mu}_i(t)$ approximation as the mean value $\hat{\mu}_i^k = \frac{\hat{\mu}_i(k\Delta T)+\hat{\mu}_i((k+1)\Delta T)}{2}$;
    run CURSA-SQ with server bitrate $\hat{\mu}_i^k$;
  end
end

Algorithm 1: Brute-Method

A first approach, algorithm 1, consists on taking K small slices of length ΔT and computing $\hat{\mu}_i$ with constant approximations in each period ΔT. Consequently, the time-dependence of the differential equations is skipped by approximating μ_i with a linear piece-wise polynomial of order 0. As a counterpart, the algorithm cannot be parallelized leading to a low performance (in time and resources), but the accuracy will be high if a fine time step is choosen. In fact, in order to have a high accuracy we need to increase the number of slices, K, decreasing the performance.

To see the differences between these models a toy example is presented in Figure 10 where three types of traffic are injected into three different queues with their corresponding server bitrate. Note that, the input traffic of the three queues is below the server bitrate and consequently the output traffic is equal to the input traffic. However, the sum of the three traffics exceed the total capacity given to these three

entities (red-dash line).

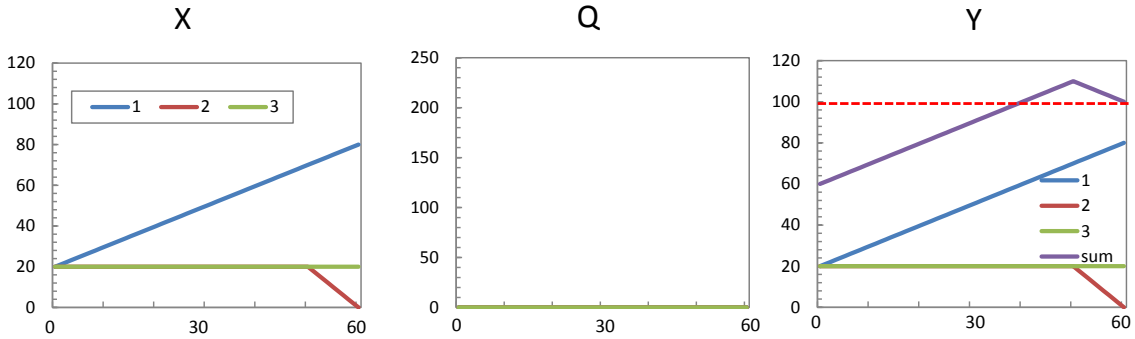

Figure 10: Example of integration for static server rate without shared medium limitation.

For the particular case of the proportional shared rate of equation 16, algortihm 1 can be improved by observing when there is excess. This allows to approximate the actual server bitrate by $\mu_{max} \cdot \frac{\mu_i}{\sum_i \mu_i}$ when there is excess ($\sum_i Y_i(t) > \mu_{max}$) and μ_i otherwise, see algorithm 2. Therefore, we only have to compute one first iteration with no medium restrictions, in order to localize the segments where there is excess, and another iteration more per segment using the uniform approximation of $\hat{\mu}_i$. So, this method ensures that the medium capacity μ_{max} is never exceeded and it is easy to implement and execute. The main drawback of this approach is the underestimation of the total output traffic, as well as the overestimation of the queued traffic.

Data: $t \in [t_0, t_{max}], Q_{0,i}, \mu_i, \mu_{max}$
Result: $Y_i(t), Q_i(t)$
for *every queue, i,* **do**
  run CURSA-SQ with server bitrate μ_i;
end
find the segments, S, in t where there is excess ($\sum_i \hat{Y}_i(t) > \mu_{max}$);
initialize $t_1 = t_0$;
for $(t_{ini}, t_{end}) \in S$ **do**
  $t_2 = t_{ini}$;
  for *every queue, i,* **do**
    run CURSA-SQ with server rate μ_i and time $t \in [t_1, t_2]$;
    $t_1 = t_{ini}$;
    $t_2 = t_{end}$;
    run CURSA-SQ with server rate $\mu_{max} \cdot \frac{\mu_i}{\sum_i \mu_i}$ and time $t \in [t_1, t_2]$;
  end
  $t_1 = t_{end}$;
end
for *every queue, i,* **do**
  run CURSA-SQ with server rate μ_i and time $t \in [t_{end}, t_{max}]$;
end

Algorithm 2: Dynamic-Bounded-Method

In Figure 11 can be seen the results for the dynamic bounded method which ensures that no limitation of capacity is exceeded but does not implies an efficient use of the shared capacity, note that the rate distribution of Figure 11 is not optimal.

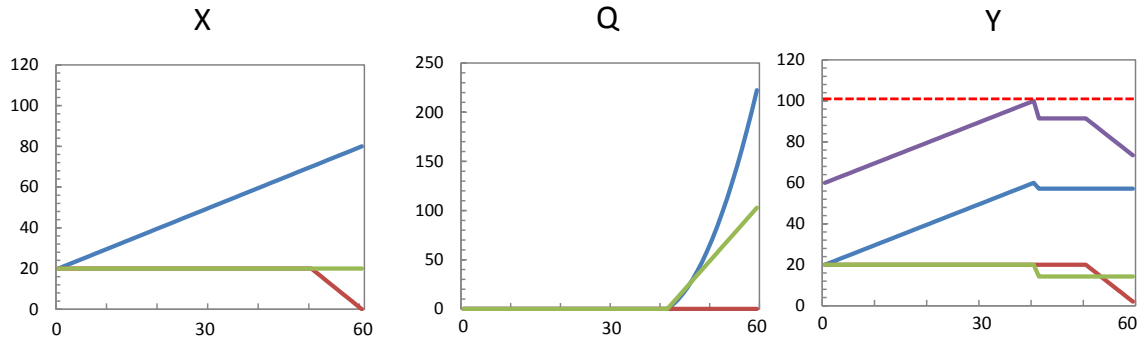

Figure 11: Characterization of the entity in terms of the queue state and the input and output traffic using the dynamic bounded approach.

Finally, if we do not use any approximation, we only have to compute the actual maximum output rate as we have seen in equation 16 and solve the non-autonomous ODE equation, having the Dynamic-Adaptative-Method. Thus, we will have to run a first CURSA-SQ per queue without medium limitation in order to compute the reduced maximum output bitrates and then solve a time-variant ODE per queue. This time-variant ODE will be the same as in the CURSA-SQ section but with a server bitrate that depends on time. The main drawback of this method is the complexity of having a non-autonomous ODE, being necessary to prove that all the properties of CURSA-SQ remains in this non-autonomous case. If we compare the two Dynamic methods the bounded one performs less accurate results but its performance is higher, using less time and resources than the adaptative one. Note that, the dynamic adaptative doubles the time needed to make an integration by the dynamic bounded method, which doubles the time needed in the static case. On the other hand, the dynamic adaptative method presents an optimal distribution of the shared capacity between the queues since all of the capacity is used but never exceeded as it can be seen in Figure 12. The Brute-Method will only be useful in order to compare with an accurate solution by using very small time-step.

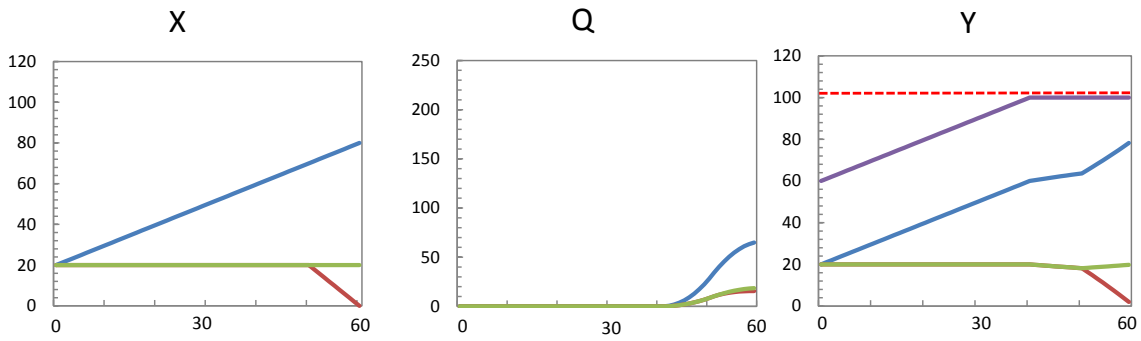

Figure 12: Characterization of the entity in terms of the queue state and the input and output traffic using the dynamic adaptative approach.

5.10 Theoretical properties

Finally, if the dynamic-adaptative integration method or any other approach that implies a non-autonomous ODE the CURSA-SQ properties for autonomous ODEs must be checked. On the other hand, the time-invariant-approximation methods (the Brute and the Dynamic Bounded) will have the same properties as CURSA-SQ method (existence, uniqueness, positivity, asymptotic behaviour and FIFO property) as it consists on repeating the CURSA-SQ methodology as many times as the number of small time-steps we considered.

Let us remark that this time-steps should be small enough to ensure that there is no excess in the capacity of the shared medium by the total traffic. This issue could be solved by computing the approximation using the minimum between the extremes points of the intervals instead of using the mean value. In the dynamic-bounded method we will have the same properties but the number of constant server bitrate intervals will be lower, and the capacity of the shared medium will never be exceeded. The main issue of these methods is that we do not profit all the capacity of the shared medium and that is why we proposed a dynamic-adaptative method.

The dynamic-adaptative method will conserve all of the CURSA-SQ properties (existence, uniqueness and positivity of the ODE solution, asymptotic behaviour and FIFO property). All of the properties are proven in a similar way as they are in [1], unless the FIFO property due to the time variant nature of the scheduled rate. That is why this property will be proven in this chapter.

Definition 5.8. Exit time. The exit time perceived by an entity i when it has arrived at time t. This time corresponds to the sum of the time which has arrived, t, and the actual delay perceived by the entity which is produced by the wait in queue, $D(t)$,

$$\Delta_i(t) = t + D(t) = t + 8\frac{Q(t)}{\mu(t)} \tag{23}$$

Proposition 5.9. FIFO property. *Let $Q(t_0) = Q_0 \geq 0$ be an initial condition, $X((t) \geq$ a continuous and positive inflow function and $\mu(t) > 0$ a maximum outflow rate. Then, if $t < s$ we have that $\Delta_i(t) < \Delta_j(s)$. In other words the exit time of an entity i that has arrived in the instant t is smaller than other j that arrives at $s > t$.*

Proof. Both exit times cannot be compared if they are not refer to the same time scale because the server bitrate given to each entity and the the delay perceived by these entities change along time, and thus the exit time. So, let us consider a time interval $T = s - t$ such that the departures (in bytes) in this period must be less than the queue state at time t, i.e. $Q(t) > 1/8 \cdot \int_t^s \mu(\xi)d\xi$; otherwise, the first entity would have already exit the queue. Then, the exit time of entity i

$$\Delta_i(t) = t + D(t) \longrightarrow \Delta_i(t+T) = t + T + 8\frac{Q(t) - 1/8 \cdot \int_t^s \mu(\xi)d\xi}{\mu(t+T)},$$

that can be compared with the exit time of entity j,

$$\Delta_j(s) = s + D(s) = s + 8\frac{Q(s)}{\mu(s)} = t + T + 8\frac{Q(t+T)}{\mu(t+T)}.$$

Impossing the FIFO property, $\Delta_i(t) < \Delta_j(s)$, we obtain

$$\frac{Q(s) - Q((t)}{s-t} > -\frac{1}{8 \cdot (s-t)} \cdot \int_t^s \mu(\xi)d\xi = -\frac{\bar{\mu}}{8} \geq -\frac{\mu_{ub}}{8}.$$

Note that the right hand side of the inequality can be bounded by the approximation of the Riemann integral,

$$\bar{\mu} = \frac{1}{(s-t)} \cdot \int_t^s \mu(\xi) d\xi \approx -\frac{1}{(s-t)} \sum_n \delta_n \mu_n,$$

where $\sum_n \delta_n = (s-t)$ and $\mu_n \geq \mu_{lb}\ \forall \xi$ because the function $\mu(\xi)$ corresponds to a bounded function. So,

$$\mu_{lb} \leq \bar{\mu} \leq \mu_{ub}.$$

Then, since $Q(t)$ is differentiable, by the *Mean-Value theorem* we know that there exists a $\xi \in (t, s)$ with $s = t + T$ such that

$$Q'(t) = \frac{Q(s) - Q(t)}{s - t}.$$

Taking into account the non-negativity property of the $\mu(t)$ function and distinguishing into the two possible cases:

- $X(\xi) < \mu$, that depends on the state of the queue:
  - If $Q(\xi) = 0$ then $Q'(\xi) = X(\xi) - X(\xi) = 0 > -\bar{\mu}$
  - If $Q(\xi) > 0$ then since the outflow is bounded by $0 < Y(\xi) < \mu(\xi)$ and the inflow is non-negative, hence
$$Q'(\xi) = \frac{X(\xi) - Y(\xi)}{8} \geq -\frac{Y(\xi)}{8} > -\frac{\mu(\xi)}{8} \geq -\frac{\mu_{ub}}{8}$$
- $X(\xi) \geq \mu$, which leads to $Q'(\xi) = (X(\xi) - \mu(\xi))/8 > -\mu_{ub}/8$

□

5.11 Conclusions

The adaptations and extensions to the original CURSA-SQ model introduced in this chapter has allowed us to use it in mobile networks where the server bitrate is not constant along time. First, the RAN model was introduced to see how a cell is discretized in terms of the services and the zones being able to configure the generators of each entity. Then, the server bitrate distributed to every entity has been presented for the particular case of having a PF scheduler. But before that, a dynamic configuration of the entities is needed since the number of devices of a given region and service change along time due to mobility. The dynamic configuration module is formed by three parts: the disaggregated traffic estimation to see how many services are in a zone and their weight in that region, the traffic projection for application purposes and the entities configuration that simplifies the topology in order to reduce the computational cost of the method. Moreover, the methodology presented in this chapter is extended by taking into account flows with different priorities, i.e. priority queues, and adding an observe-analyze-act loop with the evaluation and tunning module. Finally, we have seen some methods to integrate the ODE equation obtained from the model and we have checked that original CURSA-SQ properties are conserved despite that the resulting ODE is non-autonomous anymore.

6. Results

6.1 Introduction

Once extended the original CURSA-SQ methodology to mobile scenarios and real-time network operation we proceed to validate these extensions. First, the developed CURSA-SQ must have enough accuracy in terms of the KPIs estimation and enough time efficiency compared to the discrete-event simulators. For that reason it would be compared with the ns-3 simulator, for further information on how this discrete event simulator works visit [2].

Finally, an example of application in the field of network's operation is provided by computing the end to end KPIs of different mobility scenarios, the load measured in each layer of the example network and the disaggregation of the KPIs into the impact in the KPIs of every node of the network.

6.2 Implementation

As we have said the aim of the results is to validate the methodology and thus we compared the CURSA-SQ with the ns-3 simulator. To do that, we implemented the continuous simulator in MATLAB and we use the ns-3 open source discrete-event network simulator to compare. The MATLAB implementation consisted mainly on the modules shown in Figure 5c, having special interest the following ones:

- **Mobile Network Model**: implementation of the logistic queue model for mobile scenarios, equations 5 and 13 , by using the libraries for solving differential equations. The most common libraries are *ode45* and *ode113*. Finally, the one used was *ode45* as it is the most stable [14], but for the scenarios proven both solvers gave indistinguishable results.

  For the implementation of the scheduler module, equation 13, we need to have in mind that two steps of the CURSA-SQ must be made. The first one, with no policy $\alpha = 0$, and the second one taking into account the scheduling policy, $\alpha \neq 0$, PF in our case. Note that, the **Fixed Network Model** corresponds to considering just equation 5 with a fixed server bitrate.

- **Input traffic Characterization**: implementation of the generators, follow equations 6 and 7, where the $TC_s =< ebs, vbs, eibr, vibr >$ are the traffic characteristics of service s. Normalizing this result by the number of users of entity e we obtain the $X_s(t, TC_s)$ from equation 4 which follows the service pattern for one single user defined by the traffic characteristics. To make possible the generation of traffic the Services DB was needed to store information about the types of traffics injected to the network. In addition, the Aggregated Traffic DB and Topology DB are needed in order to know how many generators are and their configuration (percentage of of traffic of every service type and location). Both DB are given by the **Dynamic Configuration** module.

- **Initialization parameters**: in order to provide the input data to the dynamic configuration module three files are considered: one containing the service traffic characteristics and its evolution along a period of a day, the demand of the services in number of users per region and a topology of the entities which provides a mapping between the entities and the set $< \text{zone,service} >$.

6.3 CURSA-SQ vs. ns-3

In order to validate the CURSA-SQ end-to-end KPIs estimation in converged mobile-fixed scenarios, we first run a number of simulations to compare the performance of the added modules to the original CURSA-SQ methodology for shared medium and mobility scenarios against a discrete-event simulation based on the well-known ns-3 open source discrete-event network simulator [2]. In particular, we used the ns-3 LTE module [15] modeling the full LTE Radio Protocol and the Evolved Packet Core (EPC), including the core network interfaces, protocols, and entities. Both CURSA-SQ and ns-3 run on a i7-8700 server with 16GB RAM and Ubuntu 18.04.

For this comparative study, a scenario with one single base station with a three-sectored antenna, an EPC and several end-user devices receiving the traffic (UDP packets) injected by a random bursty traffic generator was simulated. Each sector was modelled as a parabolic antenna with a 3dB beam width of 70 degrees and a maximal attenuation of 20dB. For the sake of simplicity, we considered interference-free radio links with line of sight between the base station and the users. Both the UDP input traffic in terms of the peak-average ratio and the users location along the cell and the SINR map can be seen in Figure 13.

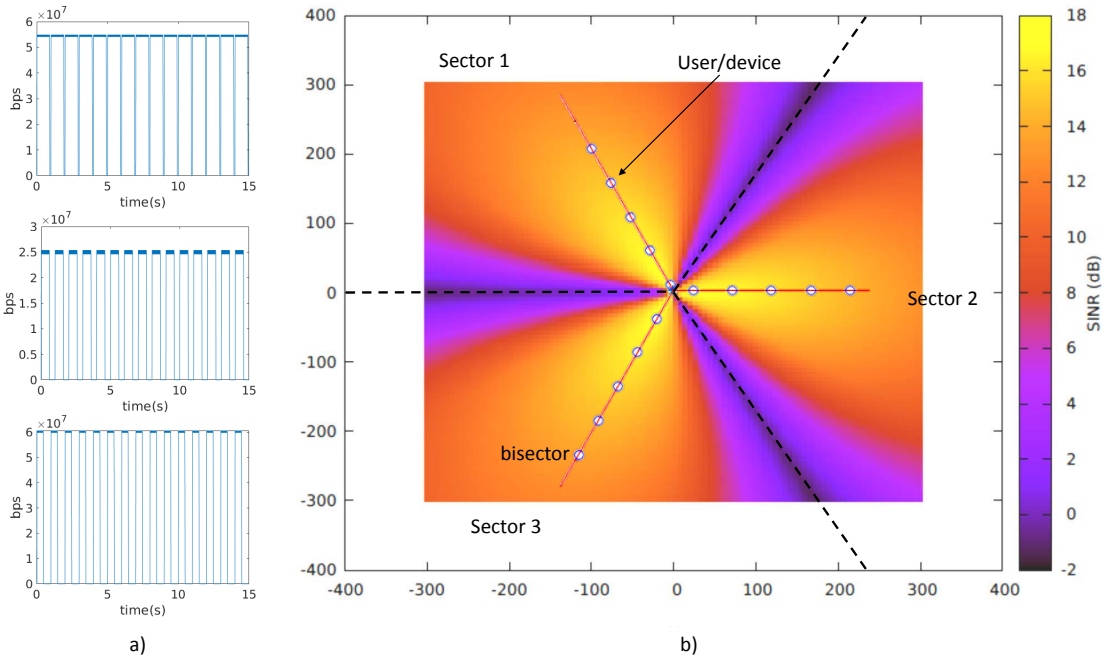

Figure 13: (a) UDP Input Traffic with different burstiness degrees (b) Users disposition and SINR map of the cell simulated.

On one hand, the ns-3 scenario was configured with the PF scheduler, 1ms transmission time interval, and 5MHz downlink bandwidth. According to the adaptive modulation and coding model in [16], the simulator finds the best Modulation and Coding Scheme (MCS) for a given channel condition. On the other hand, the PF scheduler of CURSA-SQ, equation 13, was configured taking into account the SINR map of the cell for the fixed rate term f_i and the parameters explain in Chapter 5 for the policy function $g(\cdot)$. The SINR map was provided by the ns-3 itself but could have been provided by any other implementation of the considered wave propagation model or measures. Moreover, the proportionality condition was checked

by computing the Jain Index.

Figure 14 shows the results of the simulation of 15 devices located in the bisector line of the three cell-sectors between 10 and 290 meters from the antenna. By locating the users in the bisectors we ensure that no one falls into the interference zone, i.e. very poor SINR; thus the user's SINR would be between 8 and 18 dB. The CURSA-SQ was configured with one single device per entity, i.e. 15 zones. As it can be observed, the results for relevant per-device partial KPIs, minimum and average throughput in Figure 14a) and average and maximum delay in Figure 14b), are similar for both simulation environments. The major deviations are observed for delay estimation at medium distances, where CURSA-SQ overestimates the delay; this is as a consequence of the intrinsic nature of the continuous queue model [1]. However, the impact of such overestimation is minor as they could lead to conservative decisions for the decision maker module. Remark that, since each users (located at one unique distance) generates traffic along time the results shown correspond to the minimum, mean and maximum value along time.

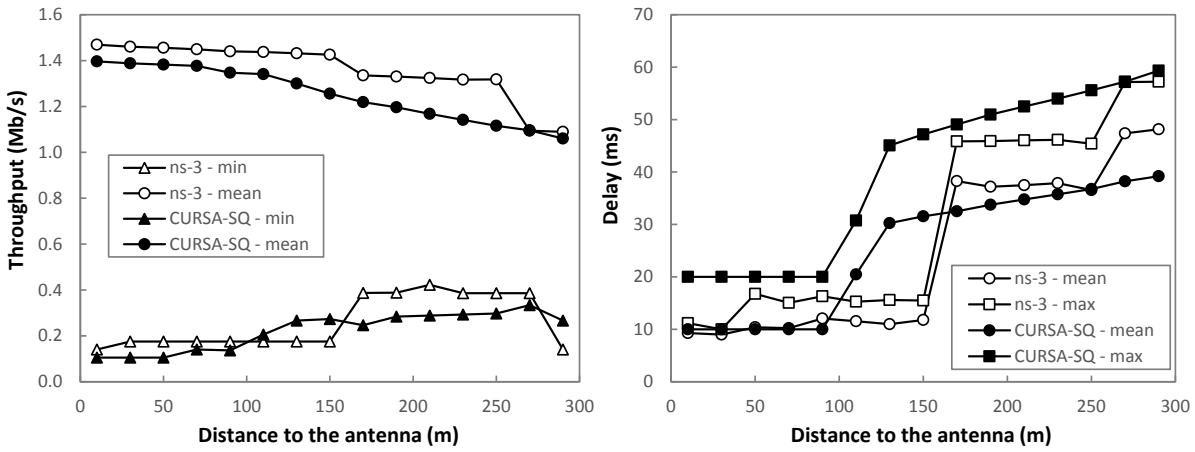

Figure 14: Throughput (a) and latency (b) in the radio segment vs distance for 15s of simulation.

Note that in the previous example the traffic generated by the users had the same characteristics: UDP with average of 1 Mbps of data and 1.5 Mbps with the signaling. The lack of change in the input traffic characteristics of every generator corresponds to one service and static scenario. So, the mobility and several services scenario must be still proven. Therefore, we made a study of the accuracy in terms of the burstiness degree of the traffic while maintaining the user's positions of the previous example. This extended comparison of the results in terms of the relative difference for estimating the relevant KPIs between ns-3 and CURSA-SQ is shown in Table 1. To variate the burstiness degree of each generated input traffic a number of repetitions with different random traffic traces and mobility patterns were simulated. Results are segmented by different peak/average traffic ratios of the traces; the higher ratio the more bursty the injected traffic. An example of the input traffic characteristics (peak-average-min) for an scenario and the SINR perceived by each user is shown in Figure 16. Note that throughput errors typically remain below 10%, whereas higher delay estimation errors are caused by the CURSA-SQ overestimation illustrated in figure 14. Recall the definition of mean relative error as

$$\varepsilon = \frac{1}{N}\sum_{i=1}^{N}\frac{X_i - \tilde{X}_i}{X_i}$$

used to compute the error between the ns-3 and CURSA-SQ of Table 1.

Table 1: CURSA-SQ vs ns-3 comparison in terms of the mean relative error, ε.

Peak-average ratio	**[1, 1.2)**	**[1.2, 1.4)**	**[1.4, 1.7)**
Throughput-min	3.5%	6.8%	10.0%
Throughput-mean	1.5%	3.7%	5.3%
Delay-mean	13.2%	14.1%	17.4%
Delay-max	20.5%	15.3%	12.9%

Moreover, in Figure 15 are shown the KPIs of a particular scenario of the scenarios simulated to compute the error between ns-3 and CURSA-SQ in Table 1. Note that, in mobility scenarios and having devices from different services in the same cell the CURSA-SQ still gives enough accurate results. In fact, looking at the results we can see that CURSA-SQ presents high accuracy in the throughput and the loss traffic estimation and an overestimation of the delay (mainly for the maximum delay).

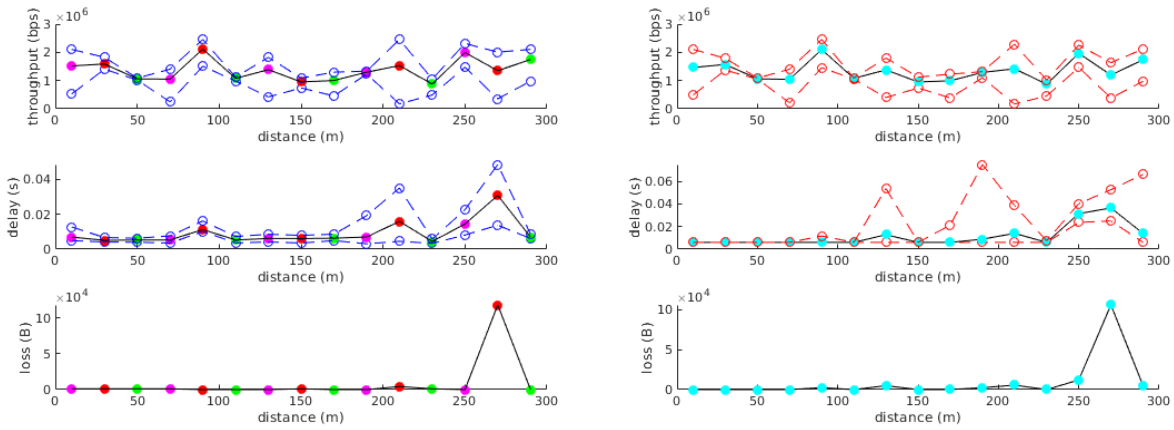

Figure 15: Estimated KPIs using ns-3 (a) and CURSA-SQ (b) for 15s of simulation. Dash line corresponds to maximum or minnimum and normal line to mean values.

To see if the implemented scheduler in MATLAB is affecting the accuracy we can plot the relation between the delay and the queue state. We have considered as queue state the sum of the queued traffic and loss normalized by the queue capacity because, then, if the queue has loss traffic the ratio is greater than one and otherwise between zero and one, Figure 16b). Note that, CURSA-SQ overestimates some of the queued traffic, i.e. overestimation of the delay, as we said. In addition, the delay and the queue state without taking into account the device which has loss traffic are linearly dependent.

The fairness introduced by the scheduler can be seen in Figure 16b); the bigger the variance is the less fair the distribution is. Then, CURSA-SQ is less fair than the ns-3 although it has a high Jain index, so the rate distribution is fair enough.

The last validation step concerns the scalability and applicability of the CURSA-SQ approach in a real-time environment, i.e. low execution (time-to-solve) time at least lower than the simulation time. To this aim, let us consider that, in order to make operational decisions and allow its implementation, simulations of 2-minute time windows need to be solved. Figure 17 shows the time-to-solve one-entity queue system model as a function of the granularity configured in CURSA-SQ. The impact of reducing the granularity

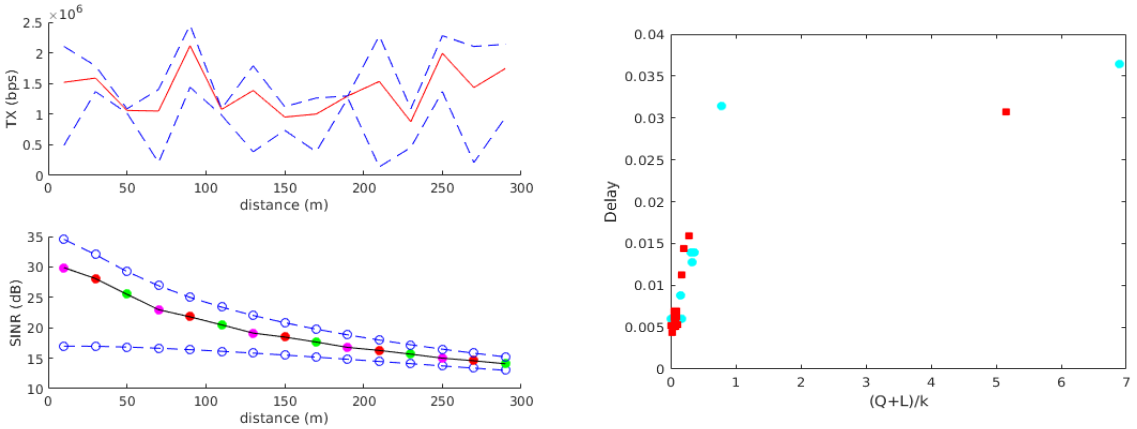

Figure 16: Input traffic of the example (a) and the delay analysis in terms of the queue state (b) for 15s of simulation. Dash line corresponds to mx. or min. and normal line to mean values.

is two-fold: while the precision of KPI estimation and the amount of information for decision making increases, the time-to-solve also increases, which can impact negatively for real-time operation. Note that, the increase of the granularity implies more evaluations of the function to solve the integral of the logistic queue model.

As it can be observed, sub-second granularities can be achieved with low execution times. For instance, if we choose the same granularity used in the accuracy comparison, 250ms, just 12.5 seconds were needed to simulate the 2-minute time-window, which is remarkably lower. Consequently, CURSA-SQ can be used for real-time operation. Note that the ns-3 simulation required 15 min thus, exceeding the simulated time-window.

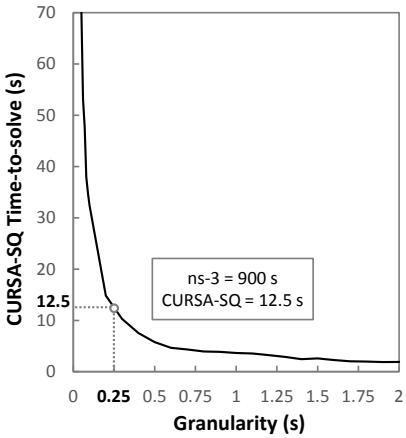

Figure 17: Time-to-solve vs granularity.

6.4 KPIs for real-time operation scenario

Let us assume the example described in Figure 18a) for the estimation of the KPIs in an advanced 4G scenario for real-time operation purposes, where every optical link has been provided with 10 Gbps and every cell capacity corresponds to four times approximately the actual 4G cells, i.e. 1 Gbps. A simple topology has been considered with only three cells, two cell side gateways (CSGw), one metro router and a core router which ends into the sink or service providers (S). In addition, some monitoring data points has been added to provide useful information for the evaluation and tunning loop and for the dynamic configuration module, see Chapter 5 for further details. Remark that, we also used background traffic in both CSGw to simulate the throughput that background cells with a normal functioning would have since having only three cells connected to two CSGw is not a realistic scenario.

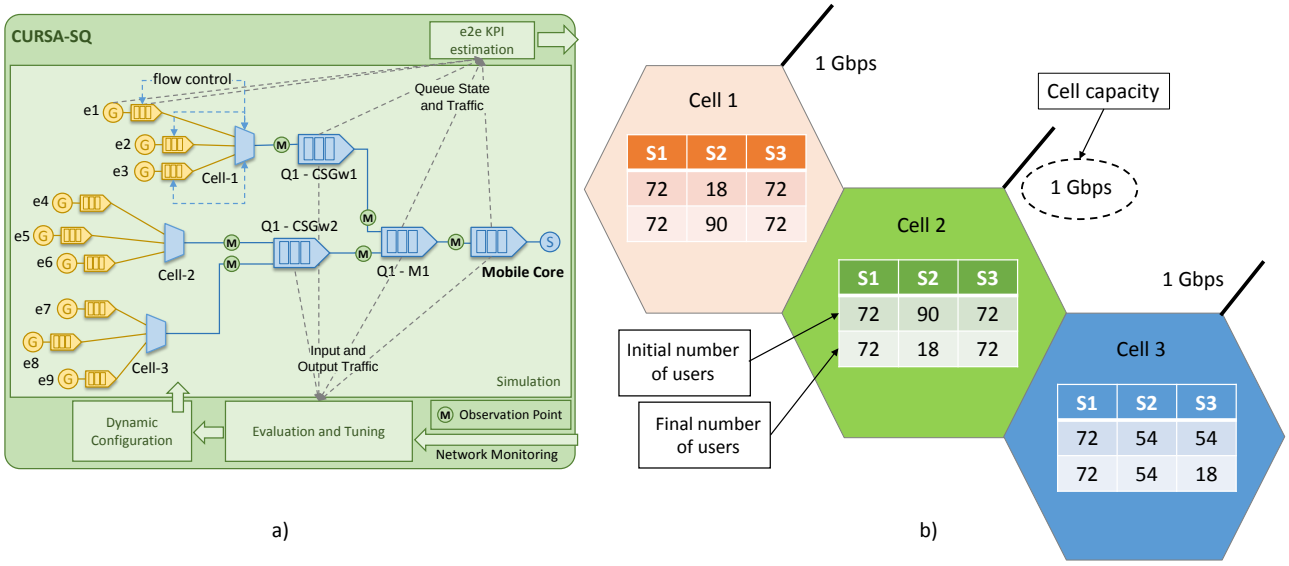

Figure 18: CURSA-SQ simulation and statistics utilization set-up (a) and its RAN configuration (b).

In order to run the simulation several parameters had to be configured. First, in the physical part of the RAN we have antennas with 20dB of maximal attenuation and cells with a radius of 100m. Then, we configured the entities topology by considering 5 zones and 3 types of services (VoD, Gaming and Internet) with the traffic characteristics shown in Table 2. Consequently, we end up with 15 entities per cell (45 in total) each one with a traffic type, a zone, a number of users profile and an allocated queue capacity.

Table 2: Service traffic characteristics.

Service	E(ibr) (s^{-1})	V(ibr) (s^{-1})	E(bs) (MB)	V(bs) (MB)
VoD	0.25	$2.54 \cdot 10^{-5}$	0.96	0.3025
Gaming	1.33	0.19	0.14	0.02
Internet	1.66	0.40	0.12	0.04

Remark that, the number of users profile has a periodicity equal to the simulation time of 2 minutes. Moreover, the demand of the services by the devices will change along the day and the week. That is why we also considered an evolution with a periodicity of a day for the service traffic generation but it has no

effect since it is much bigger than the simulated time, note that this buffer allocation mainly depends on the number of users and service provided by the entity and can change along time.

Afterwards, we set up the number of users profiles for each scenario and cell with the aim of simulating 3 types of scenarios: normal functioning (S1), delay violation (S2) and metro router congestion (S3). To reproduce these three types of scenarios we proposed a static evolution in the number of users for the first one, interchange of users between cells 1 and 2 in the second one and a external traffic connected to CSGw2 in addition to the background traffic, that simulates the growth of the throughput traffic in the background cells. Note that, in Figure 18b) the initial number and final number of active users per cell and scenario are shown. We supposed a linear evolution of users along time and a balanced-random distribution of the users among the five zones and between the three type of services. Finally, we allocated a maximum buffer size of 1.5MB for each of the entities which leads to a maximum delay of 180ms before starting to loose traffic. Recall that the entity delay corresponds to the one that suffers the entire consumer group of service type $s \in \mathcal{S}$ and zone $z \in \mathcal{Z}$ which has a given number of users $U(t)$ at instant of time $t \in [0, 2]$ minutes.

Once we run the simulation the results are post-process to show only the following KPIs:

- The values of the load in each monitoring points, computed as the throughput between the link capacity. Note that this definition of the load corresponds to a normalized throughput.
- The minimum and mean end-to-end delay for every cell.
- The minimum and mean entity load for every cell, computed as the queue state normalized by the queue capacity, which allows us to see if there are traffic loss in the shared medium layer.

Note that, all per-queue KPIs and end-to-end KPIs were also computed to check the proper functioning of the simulation but only those which provide more information about the network state were selected.

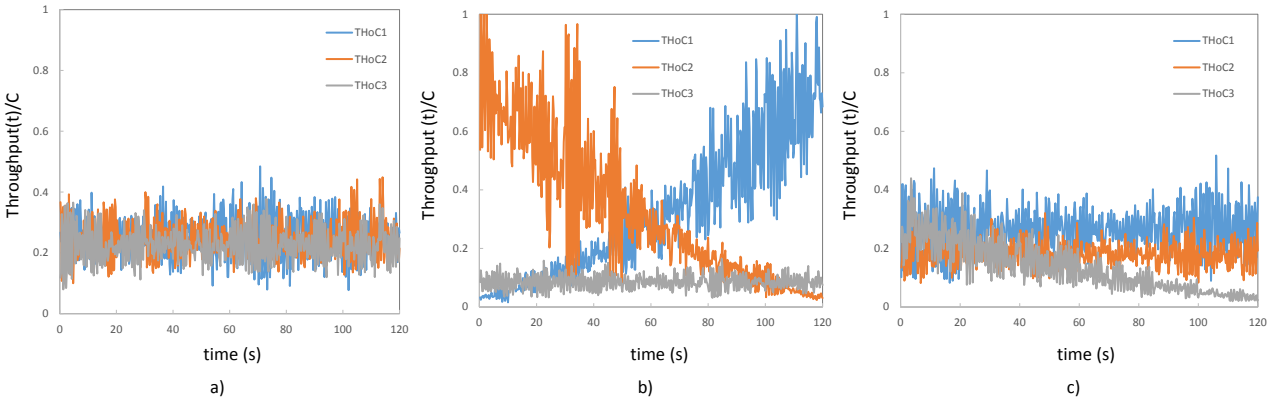

Figure 19: The load at the exit of the three cells (THoC1,THoC2 and THoC3), before entering into the CSGw for scenario S1 (a), S2 (b) and S3 (c).

Figure 19 shows the load of the cells for the three scenarios considered. In the first one there is no critical load in any of the cells since it corresponds to a normal functioning where the cell capacity is never exceeded whereas in the second scenario represents an overload of the cell capacity. Particularly, we can see that the second cell starts with a very high volume of traffic with the cell capacity overloaded but empties out along time whereas the first cell start with low traffic and starts to saturate at some time because the users of the second cell have moved to the first cell overloading it. For instance, this scenario can happen

in a demonstration where the people in it moves along the street changing of cell at a certain instant of time. Meanwhile, the users of the third cell remain static. Finally, in the third scenario which corresponds to Figure 19c) we can see that there is no risk of overloading the capacity given to each cell and the users of the first two cell remain static and the third cell empties along the time.

Once, we have analyzed the aggregated throughput for each cell having a global view of what is happening we proceed to analyze the end-to-end delay for each entity. Figure 20 shows the minimum, mean and maximum end-to-end delay and load for all the entities of cell 1 in the three scenarios considered. Note that, in the first scenario shown in Figures 20a) and 20d), the maximum per-entity delay is always bellow 80ms (around $\approx$ 20ms of per-user delay) and the entities never exceed the 50% of queue occupancy. For the second scenario, shown in Figures 20b) and 20e), the maximum delay reaches to the limitation of 180ms at time 89.25s leading to a traffic loss of the 100% during 250ms (the aggregation time or granularity) because the queue capacity is exceeded. Finally, in the third scenario, shown in Figures 20c) and 20e), we can see that the maximum entity load is even lower than in the first scenario, which seems reasonable since there are less users than in cell 1, and thus the delay would be lower than the first scenario, in particular no greater than 50ms. However, note that from time instant 100s the mean delay of the entities in cell 1 starts to increase up to 1ms and no entity load is associated to this increment. Therefore, the increase of the mean delay might be produced by one of the nodes from the fixed part of the network.

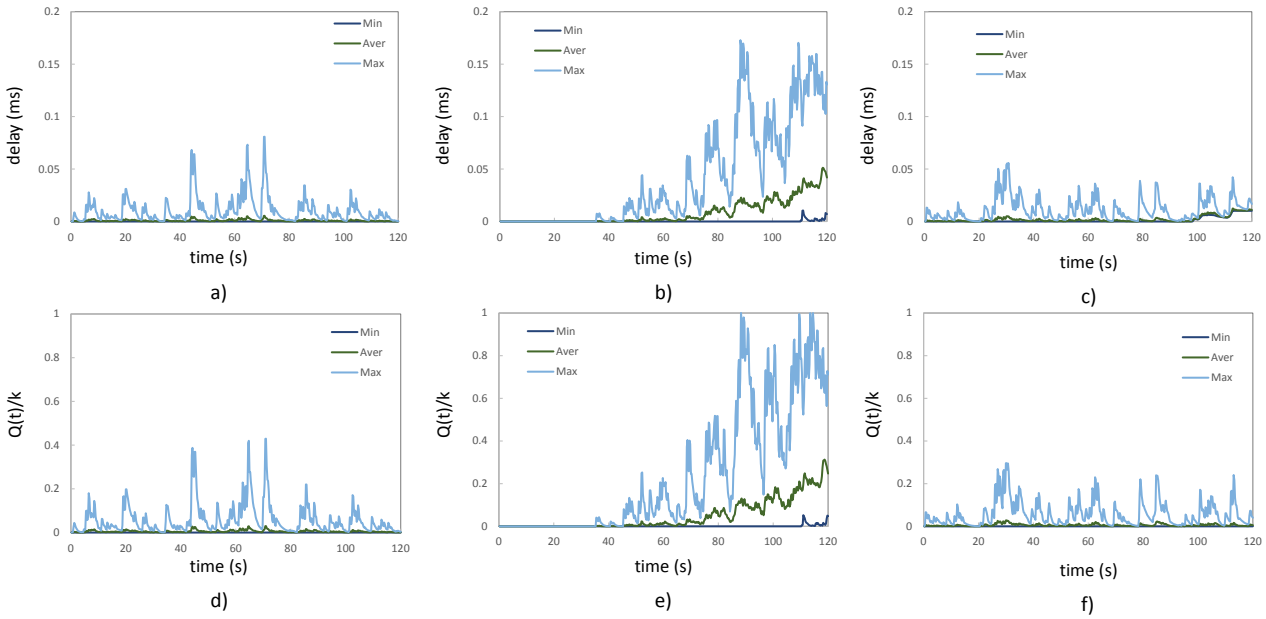

Figure 20: En-to-end delays for scenarios S1 (a), S2 (b) and S3 (c) and the entities loads for the scenarios S1 (d), S2 (e) and S3 (f).

The load can be analyzed in the rest of the topology elements considered in example of Figure 18a), as we have seen with the load of the cells. The load at the exit of both CSGw provides important information for understanding what is happening in the third scenario, Figure 21. From Figure 19c) we saw that the tendency of the aggregated throughput of cell 2 remained constant and of cell 3 decreased. But in Figure 21c) we see that the throughput of CSGw2, which is the gateway of cells 2 and 3, grows due to an increase of the traffic injected by the background cells of the network connected to CSGw2. Consequently, if the

throughput of any of the CSGw exceeds the link capacity between the gateways and the Metro router of 10Gbps the cells connected to that CSGw will suffer a delay because of the queued traffic in it. For this particular case that bound of 10 Gbps is not reached and no delay is introduced by the CSGw. Finally, on the first scenario no increase nor decrease can be appreciated in the CSGw load of both gateways whereas in the second scenario a slightly increase in the CSGw1 is appreciated due to the increase of the load of cell 1 and a slightly decrease of CSGw2 can also be appreciated due to the decrease of the load of cell 2. Note that, the impact of the increase or decrease in the cell throughput is reduced by the aggregation of throughputs in the gateways from other cells.

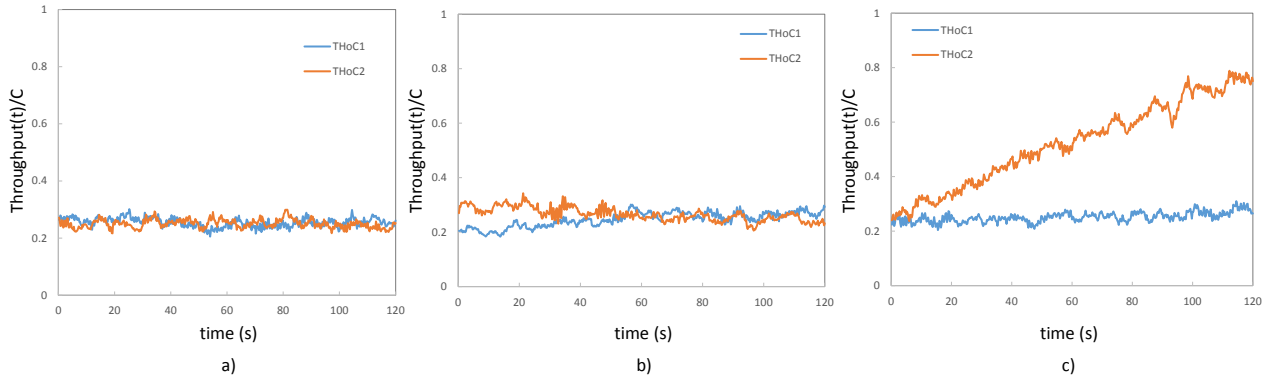

Figure 21: The load of the two CSGw for the three scenarios S1 (a), S2 (b) and S3 (c).

As we saw in the end-to-end delay results the fixed part of the network introduces a delay for the third scenario but the CSGw does not introduce any delay since their maximum capacity are not reached. So, we continue analyzing the next layer which corresponds to the metro router, Figure 22.

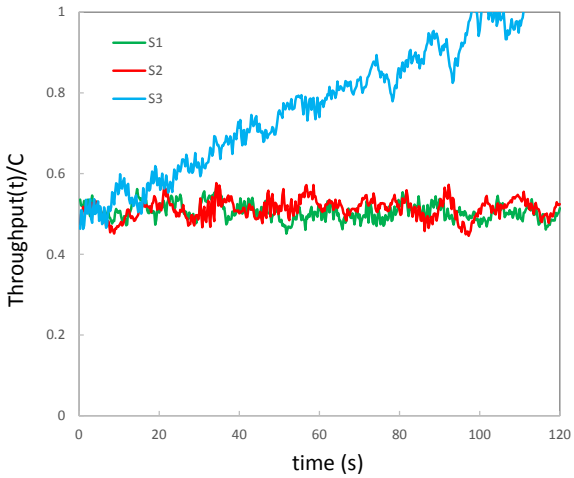

Figure 22: Load of the metro router for the three scenarios considered.

For the first two scenarios the load of the metro router is below 60% but for the third scenario the throughput reaches the maximum link capacity which means that incoming traffic starts to be backlogged introducing a delay, as we said. Note that, no traffic is lost in the metro router because we allocated enough

buffer size but if traffic continues growing there would be a point where the queue capacity is reached and traffic will be lost.

Finally, Figure 23 illustrates the usefulness of CURSA-SQ for end-to-end KPI estimation. Taking into account that the user's device does not introduce any delay and the overall numbers obtained from the previous simulations. Assuming the peak traffic of the most service-demanding user from cell 1, end-to-end delay (from device to Mobile Core) can be computed using the CURSA-SQ methodology. In fact, a better comprehension of the different scenarios is obtained. Specifically, if we have a look to the end-to-end delay of cell 1 we can distinguish between the following cases shown in Figure 23:

- normal functioning where only the scheduler introduces a normal delay of 80ms,
- delay violation where a high delay of 170ms is introduced by the scheduling process,
- the metro congestion where a part from the scheduler that introduces a delay of 32.24ms the metro router also introduces a delay of 10ms.

Note that, the metro congestion scenario can present a lower delay than a normal functioning where there are higher number of users but the metro congestion scenario is more dangerous since if the maximum queue capacity is reached by the metro router higher delay or high traffic loss could be achieve.

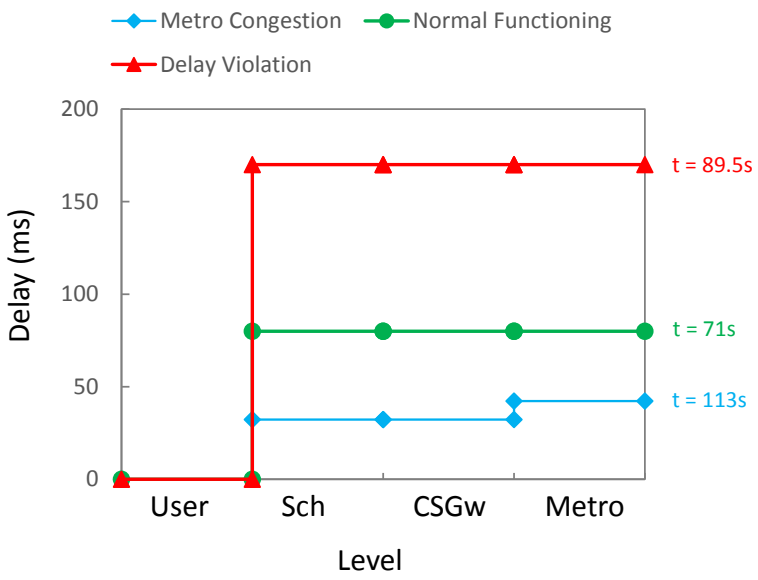

Figure 23: Components of per-user end-to-end delay for the three scenarios.

6.5 Conclusions

In this chapter we validated the extension to the CURSA-SQ methodology for shared medium and mobility enabling real-time network resource management based on current and near-term network conditions. The obtained results showed accurate, fast, and scalable end-to-end KPI estimation in converged fixed-mobile networks with sub-second granularity. In addition, this per-node KPI computation allows making precise and selective real-time operation decisions on converged access-metro environments.

7. Concluding Remarks

7.1 Contributions achieved

In this work we continued the novel fluid flow model based in the logistic queue model by extending it for mobile-fixed scenarios and real-time operation purposes. To do that, first we recalled the logistic queue model and the service-centric traffic flow generation.

Then, the KPIs were defined in order to see the necessities of the network operators and provide a field of application for these KPIs using the CURSA-SQ methodology to estimate them. Once the problematic was introduced, an extension to the CURSA-SQ methodology to estimate them was proposed. In order to extend this methodology the RAN model and the scheduling model were presented. Moreover, priority queues were also studied with the aim of generalizing the methodology. In addition, we provided some integration techniques used to solve the ODE obtained in the CURSA-SQ extended methodology and we checked that its theoretical properties were conserved.

Then, we compared the continuous fluid flow queue model with a discrete event simulator called ns-3 in order to validate the extended methodology obtaining accurate results and low execution time allowing its use for real-time operation. Finally, we provided three different scenarios where the KPIs of a converged fixed-mobile network were estimated and its impact on the network were analyzed.

This work which will be partially included as UPC contribution to the H2020-ICT-2016-2 European METRO-HAUL project (G.A. n o 761727). The main part of this work has been submitted to the IEEE/OSA Journal of Optical Communications and Networking (JOCN - 2017 IR: 2.742 - Q1 Telecommunications) with the authors Alvaro Bernal, Matias Richart, Marc Ruiz, Alberto Castro, and Luis Velasco.

7.2 Personal evaluation

During this whole year many courses from the master in mathematics (MAMME) have been useful for the elaboration of this work, for instance those regarding modelling with differential equations (Numerical Methods for Dynamical Systems) and others related with the numerical solution of them. The two courses from the master in statistics (MESIO) master I attended also gave me concepts which I applied in this work like the state space models (Time Series) used for traffic projection or statements in modelling problems (Stochastic Modelling).

In addition, this project allow me to participate in the Grup de Comunicacions Òptiques (GCO) as a research internship (GCO-AC scholarship) from September 2018 to July 2019. During this internship I improve my programming skills, mainly in MATLAB, and I learned how the network works and how can be managed and the mathematical theory behind them such as continuous queuing theory with its relation to fluid flow models. To this aim I learned some mathematical tools and how to model problems using them but also a way of thinking that allows you to isolate the problem and think of the possible solutions and the possible use cases and applications linked to this problem solution. I also have had the opportunity of collaborating with the department of radio-communications from University of Uruguay which have been very useful to understand the characteristics and functioning of radio networks and help me to obtain the results from the discrete event simulator allowing the comparison with CURSA-SQ. I also learned how to read and compare research papers, how to deal with deliverables and submissions and how to study some topics that I had not covered during my university courses.

In conclusion, it has been a very profitable and beneficial year in terms of intellectual and personal

growth although at the beginning an additional effort was needed since I came from a technical degree but the implication of the teachers from the master and the colleagues from the research group soften this beginning.

7.3 Future work

The extension to this work would be the motivation for a PhD in the field of applied mathematics for telecommunications networks. The idea would be to extend this work to more complex mobility scenarios concerning a deeper analysis of the dynamic configuration module. An impact on the network state because of the user's mobility would be provided. In addition, the methodology could be improved by taking into account more factors such as the hand-over of users between cells or more complex scenarios with interference between users obtaining not so regular patterns for the discretization of the cell.

Another extension would consist on coordinating the process of KPIs analysis with the dynamic reconfiguration of the network, so that network optimization problems can be solved given the KPI estimations, all in real-time, obtaining a better configuration of the network. For instance, the link capacity between the network nodes can change along time adapting to the network utilization. To this aim a deeper analysis of the evaluation and tunning module can be carried out, providing an integrated reconfiguration of the internal parameters of CURSA-SQ methodology and closing the observe-analyze-act loop.

Finally, other scheduling policies could be taking into account since the scheduler implemented was taken from the actual 4G networks and in 5G scenarios the scheduling policy changes. Moreover, delays can be considered in the scheduling process which leads to new scenarios where constraints over the delay can be imposed such as minimizing the jitter or ensure a known fixed delay. For instance, in M2M communication such as autonomous driving knowing the delay is crucial for the security of these applications.

References

[1] L. Velasco, M. Ruiz, and F. Coltraro. CURSA-SQ: a methodology for service-centric traffic flow analysis. *Journal of optical communications and networking*, 10(9):773–784, 2018.

[2] Open Source. Ns-3 a discrete-event network simulator. https://www.nsnam.org/docs/release/3.27/models, 2019.

[3] Ahmed Mohammed Mikaeil and Weisheng Hu. Q-learning based joint allocation of frounthaul and radio resources in multiwavelength-enabled c-ran. 2019.

[4] Almuthanna T. Nassar and Yasin Yilmaz. Reinforcement learning-based resource allocation in fog ran for iot with heterogeneous latency requirements. 2019.

[5] Richard Johnson. *Antenna Engineering Handbook (2nd ed.), p. 1-12*. McGraw-Hill, New York (NY), 1984.

[6] M. Ruiz A. Castro A. Bernal, M. Richart and L. Velasco. Modelling KPIs for 5G fixed-mobile network real-time operation. *submitted to ECOC*, 2019.

[7] Y. Xu L. Huang, B. Ding and Y. Zhou. Analysis of user behavior in a large-scale vod system. Proc. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), Taipei, Taiwan, 2017.

[8] Y. Lim D. Towsley Ch. Barakat A. Rao, A. Legout and W. Dabbous. Network characteristics of video streaming traffic. Proc. Conference on emerging Networking Experiments and Technologies (CoNEXT), Tokyo, Japan, 2011.

[9] S. R. Abied A. B. Shams and M. A. Hoque. Impact of user mobility on the performance of downlink resource scheduling in heterogeneous lte cellular networks. Proc. International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 2016.

[10] Franco Coltraro Ianniello. *A logistic queue model for network traffic modeling and simulation*. UPC commons, Barcelona, 2017.

[11] S. Ruder. An overview of gradient descent optimization algorithms.

[12] David S. Stoffer Robert H. Shumway. *Time Series Analysis and Its Applications*. Springer, Davis (CA), Pittsburgh (PA), 2010.

[13] D. Rafique and L. Velasco. Machine learning for network automation: Overview, architecture, and applications. *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, 10(9):D126–D143, 2018.

[14] MATLAB. Choose an ode solver. https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html, 2019.

[15] Open Source. Ns-3 lte module. https://www.nsnam.org/docs/models/ html/lte-design.html, 2019.

[16] M. Rossi N. Baldo M. Mezzavilla, M. Miozzo and M. Zorzi. A lightweight and accurate link abstraction model for the simulation of lte networks in ns-3. Proc. ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Paphos, Cyprus, 2012.